

Doxygen_MPD_STAR_ONLINE_DB Reference Manual

Generated by Doxygen 1.3.7

Wed Sep 6 12:03:18 2006

Contents

| | | |
|----------|---|------------|
| 1 | Doxygen_MPD_STAR_ONLINE_DB Hierarchical Index | 1 |
| 1.1 | Doxygen_MPD_STAR_ONLINE_DB Class Hierarchy | 1 |
| 2 | Doxygen_MPD_STAR_ONLINE_DB Class Index | 3 |
| 2.1 | Doxygen_MPD_STAR_ONLINE_DB Class List | 3 |
| 3 | Doxygen_MPD_STAR_ONLINE_DB File Index | 5 |
| 3.1 | Doxygen_MPD_STAR_ONLINE_DB File List | 5 |
| 4 | Doxygen_MPD_STAR_ONLINE_DB Class Documentation | 7 |
| 4.1 | tpcAnodeStateSender Class Reference | 7 |
| 4.2 | tpcAnodeStateSender_i Class Reference | 12 |
| 4.3 | tpcDewPointSender Class Reference | 16 |
| 4.4 | tpcFieldCageSender Class Reference | 26 |
| 4.5 | tpcGainMonitorSender Class Reference | 37 |
| 4.6 | tpcGasSender Class Reference | 47 |
| 4.7 | tpcISAnodeSender Class Reference | 58 |
| 4.8 | tpcISGGridSender Class Reference | 69 |
| 4.9 | tpcOSAnodeSender Class Reference | 80 |
| 4.10 | tpcOSGGridSender Class Reference | 91 |
| 5 | Doxygen_MPD_STAR_ONLINE_DB File Documentation | 103 |
| 5.1 | tpcAnodeStateDaemon.cc File Reference | 103 |
| 5.2 | tpcAnodeStateSender.cc File Reference | 104 |
| 5.3 | tpcAnodeStateSender.hh File Reference | 105 |
| 5.4 | tpcAnodeStateSender_i.cc File Reference | 106 |
| 5.5 | tpcAnodeStateSender_i.hh File Reference | 107 |
| 5.6 | tpcDewPointDaemon.cc File Reference | 108 |
| 5.7 | tpcDewPointSender.cc File Reference | 109 |
| 5.8 | tpcDewPointSender.hh File Reference | 110 |

| | | |
|------|--|-----|
| 5.9 | tpcDewPointSender_i.cc File Reference | 111 |
| 5.10 | tpcFieldCageDaemon.cc File Reference | 112 |
| 5.11 | tpcFieldCageSender.cc File Reference | 113 |
| 5.12 | tpcFieldCageSender.hh File Reference | 114 |
| 5.13 | tpcFieldCageSender_i.cc File Reference | 115 |
| 5.14 | tpcGainMonitorDaemon.cc File Reference | 116 |
| 5.15 | tpcGainMonitorSender.cc File Reference | 117 |
| 5.16 | tpcGainMonitorSender.hh File Reference | 118 |
| 5.17 | tpcGainMonitorSender_i.cc File Reference | 119 |
| 5.18 | tpcGasDaemon.cc File Reference | 120 |
| 5.19 | tpcGasSender.cc File Reference | 121 |
| 5.20 | tpcGasSender.hh File Reference | 122 |
| 5.21 | tpcGasSender_i.cc File Reference | 123 |
| 5.22 | tpcISAnodeDaemon.cc File Reference | 124 |
| 5.23 | tpcISAnodeSender.cc File Reference | 125 |
| 5.24 | tpcISAnodeSender.hh File Reference | 126 |
| 5.25 | tpcISAnodeSender_i.cc File Reference | 127 |
| 5.26 | tpcISGGridDaemon.cc File Reference | 128 |
| 5.27 | tpcISGGridSender.cc File Reference | 129 |
| 5.28 | tpcISGGridSender.hh File Reference | 130 |
| 5.29 | tpcISGGridSender_i.cc File Reference | 131 |
| 5.30 | tpcOSAnodeDaemon.cc File Reference | 132 |
| 5.31 | tpcOSAnodeSender.cc File Reference | 133 |
| 5.32 | tpcOSAnodeSender.hh File Reference | 134 |
| 5.33 | tpcOSAnodeSender_i.cc File Reference | 135 |
| 5.34 | tpcOSGGridDaemon.cc File Reference | 136 |
| 5.35 | tpcOSGGridSender.cc File Reference | 137 |
| 5.36 | tpcOSGGridSender.hh File Reference | 138 |
| 5.37 | tpcOSGGridSender_i.cc File Reference | 139 |

Chapter 1

Doxygen_MPD_STAR_ONLINE_DB Hierarchical Index

1.1 Doxygen_MPD_STAR_ONLINE_DB Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

| | |
|---------------------------------|----|
| tpcAnodeStateSender | 7 |
| tpcAnodeStateSender_i | 12 |
| tpcDewPointSender | 16 |
| tpcFieldCageSender | 26 |
| tpcGainMonitorSender | 37 |
| tpcGasSender | 47 |
| tpcISAnodeSender | 58 |
| tpcISGGridSender | 69 |
| tpcOSAnodeSender | 80 |
| tpcOSGGridSender | 91 |

Chapter 2

Doxygen_MPD_STAR_ONLINE_DB Class Index

2.1 Doxygen_MPD_STAR_ONLINE_DB Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

| | |
|---|----|
| tpcAnodeStateSender | 7 |
| tpcAnodeStateSender_i | 12 |
| tpcDewPointSender | 16 |
| tpcFieldCageSender | 26 |
| tpcGainMonitorSender | 37 |
| tpcGasSender | 47 |
| tpcISAnodeSender | 58 |
| tpcISGGridSender | 69 |
| tpcOSAnodeSender | 80 |
| tpcOSGGridSender | 91 |

Chapter 3

Doxygen_MPD_STAR_ONLINE_DB File Index

3.1 Doxygen_MPD_STAR_ONLINE_DB File List

Here is a list of all files with brief descriptions:

| | |
|---|-----|
| tpcAnodeStateDaemon.cc | 103 |
| tpcAnodeStateSender.cc | 104 |
| tpcAnodeStateSender.hh | 105 |
| tpcAnodeStateSender_i.cc | 106 |
| tpcAnodeStateSender_i.hh | 107 |
| tpcDewPointDaemon.cc | 108 |
| tpcDewPointSender.cc | 109 |
| tpcDewPointSender.hh | 110 |
| tpcDewPointSender_i.cc | 111 |
| tpcFieldCageDaemon.cc | 112 |
| tpcFieldCageSender.cc | 113 |
| tpcFieldCageSender.hh | 114 |
| tpcFieldCageSender_i.cc | 115 |
| tpcGainMonitorDaemon.cc | 116 |
| tpcGainMonitorSender.cc | 117 |
| tpcGainMonitorSender.hh | 118 |
| tpcGainMonitorSender_i.cc | 119 |
| tpcGasDaemon.cc | 120 |
| tpcGasSender.cc | 121 |
| tpcGasSender.hh | 122 |
| tpcGasSender_i.cc | 123 |
| tpcISAnodeDaemon.cc | 124 |
| tpcISAnodeSender.cc | 125 |
| tpcISAnodeSender.hh | 126 |
| tpcISAnodeSender_i.cc | 127 |
| tpcISGGridDaemon.cc | 128 |
| tpcISGGridSender.cc | 129 |
| tpcISGGridSender.hh | 130 |
| tpcISGGridSender_i.cc | 131 |
| tpcOSAnodeDaemon.cc | 132 |
| tpcOSAnodeSender.cc | 133 |

| | |
|---------------------------------------|-----|
| tpcOSAnodeSender.hh | 134 |
| tpcOSAnodeSender_i.cc | 135 |
| tpcOSGGridDaemon.cc | 136 |
| tpcOSGGridSender.cc | 137 |
| tpcOSGGridSender.hh | 138 |
| tpcOSGGridSender_i.cc | 139 |

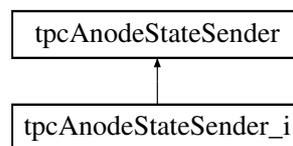
Chapter 4

Doxygen_MPD_STAR_ONLINE_DB Class Documentation

4.1 tpcAnodeStateSender Class Reference

```
#include <tpcAnodeStateSender.hh>
```

Inheritance diagram for tpcAnodeStateSender::



Public Member Functions

- [tpcAnodeStateSender](#) (const char *localDir)
- virtual [~tpcAnodeStateSender](#) ()
- virtual void [initTable](#) ()
- virtual void [initTags](#) ()=0
- virtual void [initDataBase](#) ()=0
- virtual bool [loadUserControls](#) (const char *name, const char *value)=0
- virtual void [initQuery](#) ()=0
- virtual bool [queryData](#) ()=0
- virtual bool [readData](#) (const char *fileName)
- virtual bool [updateDb](#) (const char *fileName)
- virtual bool [readData](#) (ifstream &from)=0
- virtual bool [hasChanged](#) (int rowNumber)=0

Protected Attributes

- tpcAnodeState [previousVals](#) [NUM_DB_ROWS]
- tpcAnodeState [tempVals](#) [NUM_DB_ROWS]

- int [elementList](#) [NUM_DB_ROWS]
- [tpcAnodeState](#) [updateVals](#) [NUM_DB_ROWS]
- int [updateElements](#) [NUM_DB_ROWS]
- bool [mreadStatus](#)

4.1.1 Constructor & Destructor Documentation

4.1.1.1 [tpcAnodeStateSender::tpcAnodeStateSender](#) (const char * *localDir*)

Definition at line 18 of file [tpcAnodeStateSender.cc](#).

```
18 { }
```

4.1.1.2 virtual [tpcAnodeStateSender::~~tpcAnodeStateSender](#) () [inline, virtual]

Definition at line 33 of file [tpcAnodeStateSender.hh](#).

```
33 {};
```

4.1.2 Member Function Documentation

4.1.2.1 virtual bool [tpcAnodeStateSender::hasChanged](#) (int *rowNumber*) [pure virtual]

Implemented in [tpcAnodeStateSender_i](#).

4.1.2.2 virtual void [tpcAnodeStateSender::initDataBase](#) () [pure virtual]

Implemented in [tpcAnodeStateSender_i](#).

4.1.2.3 virtual void [tpcAnodeStateSender::initQuery](#) () [pure virtual]

Implemented in [tpcAnodeStateSender_i](#).

4.1.2.4 void [tpcAnodeStateSender::initTable](#) () [virtual]

Definition at line 21 of file [tpcAnodeStateSender.cc](#).

```
21                                     {
22 #define __METHOD__ "initTable()"
23
24     StDbTable* table=0;
25     if(!(table=node->addDbTable("tpcAnodeState")))
26         sendMess("Could not find table=tpcAnodeState",dbMFatal,__LINE__,__CLASS__,__METHOD__);
27
28     memset(tempVals,0,NUM_DB_ROWS*sizeof(tpcAnodeState));
29     memset(previousVals,0,NUM_DB_ROWS*sizeof(tpcAnodeState));
30
31     int nrows;
32     int* elist = table->getElementID(nrows);
33     if(nrows!=NUM_DB_ROWS){
```

```

34     //char mess[256];
35     ostreamstream ms;
36     ms<<"Db rows("&<<nrows<<" != compiled("<<NUM_DB_ROWS<<" "<<ends;
37     sendMess((ms.str()).c_str(),dbmFatal, __LINE__, __CLASS__, __METHOD__);
38 }
39 memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
40
41 unsigned int timestamp=time(NULL);
42 mgr->setRequestTime(timestamp);
43 if(mgr->fetchDbTable(table)){
44     tpcAnodeState* thv = (tpcAnodeState*)table->GetTable();
45     memcpy(previousVals,thv,nrows*sizeof(tpcAnodeState));
46 }
47
48 #undef __METHOD__
49 };

```

4.1.2.5 virtual void tpcAnodeStateSender::initTags () [pure virtual]

Implemented in [tpcAnodeStateSender_i](#).

4.1.2.6 virtual bool tpcAnodeStateSender::loadUserControls (const char * name, const char * value) [pure virtual]

Implemented in [tpcAnodeStateSender_i](#).

4.1.2.7 virtual bool tpcAnodeStateSender::queryData () [pure virtual]

Implemented in [tpcAnodeStateSender_i](#).

4.1.2.8 virtual bool tpcAnodeStateSender::readData (ifstream & from) [pure virtual]

Implemented in [tpcAnodeStateSender_i](#).

4.1.2.9 bool tpcAnodeStateSender::readData (const char * fileName) [virtual]

Definition at line 52 of file tpcAnodeStateSender.cc.

```

52                                     {
53 #define __METHOD__ "readData(fileName)"
54
55     ifstream from(fileName);
56     if(!from) return sendMess("Cannot open file=",fileName,dbmErr, __LINE__, __CLASS__, __METHOD__);
57
58     return readData(from); // user implemented file read
59 #undef __METHOD__
60 }

```

4.1.2.10 bool tpcAnodeStateSender::updateDb (const char * fileName) [virtual]

Definition at line 63 of file tpcAnodeStateSender.cc.

```

63                                     {
64 #define __METHOD__ "updateDb(filename)"
65
66 if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
67
68 int* elements;
69 tpcAnodeState* vals;
70 int numRows = 0;
71
72 if(writeRequired()){
73     numRows=NUM_DB_ROWS;
74     elements=elementList;
75     vals = tempVals;
76
77 } else {
78     for(int i=0; i<NUM_DB_ROWS; i++){
79         if(hasChanged(i)){
80             updateElements[numRows]=elementList[i];
81             updateVals[numRows] = tempVals[i];
82             previousVals[i]=tempVals[i];
83             numRows++;
84         }
85     }
86     elements = updateElements;
87     vals      = updateVals;
88 }
89
90 if(numRows==0) return sendMess(" No update required for",mbaseName,dbMDebug,__LINE__,__CLASS__,__METHOD__);
91
92 //char mess[256];
93 ostream sn;
94 sn<<"Will Update "<<numRows<<" of "<<NUM_DB_ROWS<<" rows "<<ends;
95 sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
96
97 StDbTable* dbTable=node->findTable("tpcAnodeState");
98 dbTable->SetTable((char*)vals, numRows, elements);
99 mgr->setStoreTime(writeTime);
100
101 if(!mgr->storeDbTable(dbTable)) {
102     addBackLog(writeTime);
103     return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
104 }
105
106 if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
107
108 return true;
109 #undef __METHOD__
110 }

```

4.1.3 Member Data Documentation

4.1.3.1 int [tpcAnodeStateSender::elementList](#)[NUM_DB_ROWS] [protected]

Definition at line 24 of file tpcAnodeStateSender.hh.

4.1.3.2 bool [tpcAnodeStateSender::mreadStatus](#) [protected]

Definition at line 28 of file tpcAnodeStateSender.hh.

4.1.3.3 tpcAnodeState [tpcAnodeStateSender::previousVals](#)[NUM_DB_ROWS] [protected]

Definition at line 22 of file tpcAnodeStateSender.hh.

4.1.3.4 tpcAnodeState [tpcAnodeStateSender::tempVals](#)[NUM_DB_ROWS] [protected]

Definition at line 23 of file tpcAnodeStateSender.hh.

4.1.3.5 int [tpcAnodeStateSender::updateElements](#)[NUM_DB_ROWS] [protected]

Definition at line 26 of file tpcAnodeStateSender.hh.

4.1.3.6 tpcAnodeState [tpcAnodeStateSender::updateVals](#)[NUM_DB_ROWS] [protected]

Definition at line 25 of file tpcAnodeStateSender.hh.

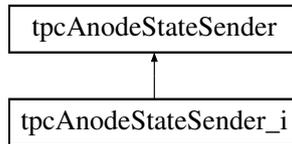
The documentation for this class was generated from the following files:

- [tpcAnodeStateSender.hh](#)
- [tpcAnodeStateSender.cc](#)

4.2 tpcAnodeStateSender_i Class Reference

```
#include <tpcAnodeStateSender_i.hh>
```

Inheritance diagram for tpcAnodeStateSender_i::



Public Member Functions

- [tpcAnodeStateSender_i](#) (const char *localDir)
- virtual [~tpcAnodeStateSender_i](#) ()
- virtual void [initTags](#) ()
- virtual void [initDataBase](#) ()
- virtual bool [loadUserControls](#) (const char *name, const char *value)
- virtual void [initQuery](#) ()
- virtual bool [queryData](#) ()
- virtual bool [readData](#) (ifstream &from)
- virtual bool [hasChanged](#) (int rowNumber)

4.2.1 Constructor & Destructor Documentation

4.2.1.1 tpcAnodeStateSender_i::tpcAnodeStateSender_i (const char * localDir)

Definition at line 17 of file tpcAnodeStateSender_i.cc.

```

17                                     : tpcAnodeStateSender(localDir){
18
19     // several layers of init -> some in baseclass, some in user edited file
20     initTags();
21     if(localDir) cd(localDir);// note this ignores the sub dir tag
22     init("tpcAnodeState"); // setup the file I/O
23     initDataBase();          // database connections
24     initTable();            // table definitions
25
26 };
  
```

4.2.1.2 virtual tpcAnodeStateSender_i::~~tpcAnodeStateSender_i () [inline, virtual]

Definition at line 26 of file tpcAnodeStateSender_i.hh.

```
26 {};
```

4.2.2 Member Function Documentation

4.2.2.1 bool tpcAnodeStateSender_i::hasChanged (int *rowNumber*) [virtual]

Implements [tpcAnodeStateSender](#).

Definition at line 141 of file tpcAnodeStateSender_i.cc.

```

141                                     {
142
143 tpcAnodeState* pre=&previousVals[rowNumber];
144 tpcAnodeState* cur=&tempVals[rowNumber];
145
146 if(pre->ARCNetOuterSector != cur->ARCNetOuterSector)return true;
147 if(pre->ARCNetInnerSector != cur->ARCNetInnerSector)return true;
148 if(pre->HVOuterSector != cur->HVOuterSector)return true;
149 if(pre->HVInnerSector != cur->HVInnerSector)return true;
150
151 /* example ... note -> change to any element requires db-update
152 * and thus returns true immediately
153 *
154 *if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
155 *if(fabs(pre->chlVoltage-cur->chlVoltage)>=driftLimit) return true;
156 *
157 * ....
158 */
159
160 return false;
161 }
```

4.2.2.2 void tpcAnodeStateSender_i::initDataBase () [virtual]

Implements [tpcAnodeStateSender](#).

Definition at line 44 of file tpcAnodeStateSender_i.cc.

```

44                                     {
45 #define __METHOD__ "initDataBase()"
46
47 /* More than an example... swap user & dbTrg as per subsystem*/
48 mgr->setUser("startpc","");
49 StDbType dbT = dbConditions;
50 StDbDomain dbD = dbTpc;
51
52 if( !( node = mgr->initConfig(dbT,dbD) )
53     sendMess("Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal,__LINE__,__CLASS__,__METHOD__);
54
55 #undef __METHOD__
56 }
```

4.2.2.3 void tpcAnodeStateSender_i::initQuery () [virtual]

Implements [tpcAnodeStateSender](#).

Definition at line 80 of file tpcAnodeStateSender_i.cc.

```

80                                     {
81
82     ofstream to(queryFile);
```

```

83
84     memset(tempVals,0,NUM_DB_ROWS*sizeof(tpcAnodeState));
85
86     to<<"tpchv:ARCNET_0"<<endl; // outer-sector arcnet
87     to<<"tpchv:ARCNET_1"<<endl; // inner-sector annod
88     to <<"tpchv:CHK_SYS_0.E" << endl; //Crate HV Status On if not zeroc
89     to <<"tpchv:CHK_SYS_1.E" << endl; //Crate HV Status On if not zero
90
91     to.close();
92 }

```

4.2.2.4 void tpcAnodeStateSender_i::initTags () [virtual]

Implements [tpcAnodeStateSender](#).

Definition at line 33 of file tpcAnodeStateSender_i.cc.

```

33                                     {
34     /* more than an example -> swap "trg" to your subsys & add to email list*/
35     setEmailTo("porter@bnl.gov");
36     setDomainName("tpc");
37
38 }

```

4.2.2.5 bool tpcAnodeStateSender_i::loadUserControls (const char * name, const char * value) [virtual]

Implements [tpcAnodeStateSender](#).

Definition at line 62 of file tpcAnodeStateSender_i.cc.

```

62                                     {
63 #define __METHOD__ "loadUserControls(name,value)"
64
65 /* more than an example ... swap driftLimit to yours
66 * and duplicate this structure for each selection criteria
67 if(strstr(name,"driftLimit")){
68     driftLimit=atof(value);
69     sendMess("driftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
70     return true;
71 }
72 */
73
74 return false;
75 #undef __METHOD__
76 }

```

4.2.2.6 bool tpcAnodeStateSender_i::queryData () [virtual]

Implements [tpcAnodeStateSender](#).

Definition at line 96 of file tpcAnodeStateSender_i.cc.

```

96                                     {
97 #define __METHOD__ "queryData()"
98
99     /*

```

```
100  * MORE THAN AN EXAMPLE...
101  * IF Standard SC-Query via "caGet" then,
102  * no need to change this method AT ALL
103  *
104  */
105
106  writeTime = (unsigned int)time(NULL);      //for database write time
107
108  char systemCmd[1024];
109  ostrstream scmd(systemCmd,1024);
110  scmd<<"caGet " <<queryFile<<" " <<dataFile<<ends;
111
112  if(system(systemCmd))
113      return sendMess(" caGet system call returned error",dbMErr, __LINE__, __CLASS__, __METHOD__);
114
115  return true;
116 #undef __METHOD__
117 };
```

4.2.2.7 bool tpcAnodeStateSender_i::readData (ifstream & from) [virtual]

Implements [tpcAnodeStateSender](#).

Definition at line 122 of file tpcAnodeStateSender_i.cc.

```
122                                     {
123 #define __METHOD__ "readData(ifstream)"
124
125 string bar;
126 mreadStatus=true;
127
128
129 from>>bar>>tempVals[0].ARCNetOuterSector; // if(!from)clearAndNotify(from,bar);
130 from>>bar>>tempVals[0].ARCNetInnerSector; // if(!from)clearAndNotify(from,bar);
131 from>>bar>>tempVals[0].HVOuterSector; // if(!from)clearAndNotify(from,bar);
132 from>>bar>>tempVals[0].HVInnerSector; // if(!from)clearAndNotify(from,bar);
133
134 from.close();
135 return true;
136 #undef __METHOD__
137 }
```

The documentation for this class was generated from the following files:

- [tpcAnodeStateSender_i.hh](#)
- [tpcAnodeStateSender_i.cc](#)

4.3 tpcDewPointSender Class Reference

```
#include <tpcDewPointSender.hh>
```

Public Member Functions

- [tpcDewPointSender](#) (const char *localDir)
- virtual [~tpcDewPointSender](#) ()
- virtual void [initTable](#) ()
- virtual void [initTags](#) ()
- virtual void [initDataBase](#) ()
- virtual bool [loadUserControls](#) (const char *name, const char *value)
- virtual void [initQuery](#) ()
- virtual bool [queryData](#) ()
- virtual bool [readData](#) (const char *fileName)
- virtual bool [updateDb](#) (const char *fileName)
- virtual bool [readData](#) (ifstream &from)
- virtual bool [hasChanged](#) (int rowNumber)
- char * [readAny](#) ()
- bool [readVal](#) (char *&value)
- bool [readVal](#) (float &value)
- bool [readVal](#) (double &value)
- bool [readVal](#) (short &value)
- bool [readVal](#) (int &value)
- bool [readVal](#) (long &value)
- bool [readVal](#) (long long &value)
- bool [nextLine](#) (ifstream &from)
- void [readError](#) (int l, char *c, char *m)

Protected Attributes

- tpcDewPoint [previousVals](#) [NUM_DB_ROWS]
- tpcDewPoint [tempVals](#) [NUM_DB_ROWS]
- int [elementList](#) [NUM_DB_ROWS]
- tpcDewPoint [updateVals](#) [NUM_DB_ROWS]
- int [updateElements](#) [NUM_DB_ROWS]
- bool [mreadStatus](#)
- char [mline](#) [256]
- char [tmpline](#) [256]
- char * [ptr1](#)
- char * [ptr2](#)
- float [hdriftLimit](#)
ditto
- float [tdriftLimit](#)

4.3.1 Constructor & Destructor Documentation

4.3.1.1 tpcDewPointSender::tpcDewPointSender (const char * *localDir*)

Definition at line 19 of file tpcDewPointSender.cc.

```

19                                     {
20
21     initTags();
22     if(localDir) cd(localDir);// note this ignores the sub dir tag
23     init("tpcDewPoint"); // setup the file I/O
24     initDataBase();        // database connections
25     initTable();           // table definitions
26
27 }
```

4.3.1.2 virtual tpcDewPointSender::~~tpcDewPointSender () [inline, virtual]

Definition at line 43 of file tpcDewPointSender.hh.

```
43 {};
```

4.3.2 Member Function Documentation

4.3.2.1 bool tpcDewPointSender::hasChanged (int *rowNumber*) [virtual]

Definition at line 114 of file tpcDewPointSender_i.cc.

```

114                                     {
115
116     tpcDewPoint* pre=&previousVals[rowNumber];
117     tpcDewPoint* cur=&tempVals[rowNumber];
118
119     if(fabs(pre->humidity-cur->humidity)>=hdriftLimit)return true;
120     if(fabs(pre->fTemperature-cur->fTemperature)>=tdriftLimit)return true;
121     if(fabs(pre->fDewPoint-cur->fDewPoint)>=tdriftLimit)return true;
122
123     /* example ... note -> change to any element requires db-update
124      * and thus returns true immediately
125      *
126      *if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
127      *if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
128      *
129      * ....
130      */
131
132     return false;
133 }
```

4.3.2.2 void tpcDewPointSender::initDataBase () [virtual]

Definition at line 75 of file tpcDewPointSender.cc.

```

75                                     {
76 #define __METHOD__ "initDataBase()"
```

```

77
78  /* More than an example... swap user & dbTrg as per subsystem*/
79  mgr->setUser("stardb","");
80  StDbType   dbT = dbConditions;
81  StDbDomain dbD = dbTpc;
82
83  if( !( node = mgr->initConfig(dbT,dbD) ) )
84      sendMess("Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal, __LINE__, __CLASS__, __METHOD__);
85
86 #undef __METHOD__
87 }

```

4.3.2.3 void tpcDewPointSender::initQuery () [virtual]

Definition at line 49 of file tpcDewPointSender_i.cc.

```

49                                     {
50 #define __METHOD__ "initQuery()"
51
52     ofstream to(queryFile);
53
54     if(!to.is_open()){
55         sendMess("Open Failed ",queryFile,dbMFatal, __LINE__, __CLASS__, __METHOD__);
56         return;
57     }
58
59     to << "dew_val1" << endl;
60     to << "dew_val2" << endl;
61     to << "dew_val3" << endl;
62     to << "dew_val4" << endl;
63     to << "dew_val5" << endl;
64
65  /* example
66  *   for(int i=0;i<16;i++){
67  *       to<<"TRGhv:SUB_RD_V_1:"<<i<<".E"<<endl;
68  *       to<<"TRGhv:SUB_RD_V_1:"<<i<<".F"<<endl;
69  *       ....
70  *   }
71  */
72
73
74     to.close();
75
76 #undef __METHOD__
77 }

```

4.3.2.4 void tpcDewPointSender::initTable () [virtual]

Definition at line 30 of file tpcDewPointSender.cc.

```

30                                     {
31 #define __METHOD__ "initTable()"
32
33     StDbTable* table=0;
34     if(!(table=node->addDbTable("tpcDewPoint")))
35         sendMess("Could not find table=tpcDewPoint",dbMFatal, __LINE__, __CLASS__, __METHOD__);
36
37     memset(tempVals,0,NUM_DB_ROWS*sizeof(tpcDewPoint));
38     memset(previousVals,0,NUM_DB_ROWS*sizeof(tpcDewPoint));
39
40     int nrows;

```

```

41     int* elist = table->getElementID(nrows);
42     if(nrows!=NUM_DB_ROWS){
43         // char mess[256];
44         ostringstream ms;
45         ms<<"Db rows("<<nrows<<" ) != compiled("<<NUM_DB_ROWS<<")";
46         sendMess((ms.str()).c_str(),dbMFatal,__LINE__,__CLASS__,__METHOD__);
47     }
48     memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
49
50     unsigned int timestamp=time(NULL);
51     mgr->setRequestTime(timestamp);
52     if(mgr->fetchDbTable(table)){
53         tpcDewPoint* thv = (tpcDewPoint*)table->GetTable();
54         memcpy(previousVals,thv,nrows*sizeof(tpcDewPoint));
55     }
56
57 #undef __METHOD__
58 };

```

4.3.2.5 void tpcDewPointSender::initTags () [virtual]

Definition at line 66 of file tpcDewPointSender.cc.

```

66     {
67     /* more than an example -> swap "trg" to your subsys & add to email list*/
68     setEmailTo("porter@bnl.gov");
69     setDomainName("tpc");
70
71 }

```

4.3.2.6 bool tpcDewPointSender::loadUserControls (const char * name, const char * value) [virtual]

Definition at line 20 of file tpcDewPointSender_i.cc.

```

20     {
21 #define __METHOD__ "loadUserControls(name,value)"
22
23 /* more than an example ... swap driftLimit to yours
24 * and duplicate this structure for each selection criteria
25 if(strstr(name,"driftLimit")){
26     driftLimit=atof(value);
27     sendMess("driftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
28     return true;
29 }
30 */
31 if(strstr(name,"hdriftLimit")){
32     hdriftLimit=atof(value);
33     sendMess("hdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
34     return true;
35 }
36
37 if(strstr(name,"tdriftLimit")){
38     tdriftLimit=atof(value);
39     sendMess("tdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
40     return true;
41 }
42
43 return false;
44 #undef __METHOD__
45 }

```

4.3.2.7 bool tpcDewPointSender::nextLine (ifstream &from) [inline]

Definition at line 76 of file tpcDewPointSender.hh.

```

76                                     {
77   if(!from.getline(mline,255))return false;
78   return true;
79 }
```

4.3.2.8 bool tpcDewPointSender::queryData () [virtual]

Definition at line 91 of file tpcDewPointSender.cc.

```

91                                     {
92 #define __METHOD__ "queryData()"
93
94 /*
95  * MORE THAN AN EXAMPLE...
96  * IF Standard SC-Query via "caGet" then,
97  * no need to change this method AT ALL
98  *
99  */
100
101 writeTime = (unsigned int)time(NULL); //for database write time
102
103 //char systemCmd[1024];
104 ostringstream scmd;
105 scmd<<"caGet "<<queryFile<<" "<<dataFile;
106
107 if(system((scmd.str()).c_str()))
108     return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
109
110 return true;
111 #undef __METHOD__
112 };
```

4.3.2.9 char * tpcDewPointSender::readAny ()

Definition at line 197 of file tpcDewPointSender.cc.

```

197                                     {
198
199   strcpy(tmpline,mline);
200   ptr1=tmpline;
201   ptr2=strtok(ptr1," ");
202   if(!ptr2) return ptr2;
203   ptr2=strtok(NULL," ");
204   return ptr2;
205 }
```

4.3.2.10 bool tpcDewPointSender::readData (ifstream &from) [virtual]

Definition at line 82 of file tpcDewPointSender_i.cc.

```

82                                     {
83 #define __METHOD__ "readData(ifstream)"
```

```

84
85 mreadStatus=true;
86 memset(tempVals,0,NUM_DB_ROWS*sizeof(tpcDewPoint));
87
88
89 int i=0;
90 char* c=__CLASS__;
91 char* m=__METHOD__;
92
93 if(!nextLine(from) || !readVal(tempVals[0].humidity)) readError(__LINE__,c,m);
94 if(!nextLine(from) || !readVal(tempVals[0].fTemperature)) readError(__LINE__,c,m);
95 if(!nextLine(from) || !readVal(tempVals[0].cTemperature)) readError(__LINE__,c,m);
96 if(!nextLine(from) || !readVal(tempVals[0].fDewPoint)) readError(__LINE__,c,m);
97 if(!nextLine(from) || !readVal(tempVals[0].cDewPoint)) readError(__LINE__,c,m);
98 /* example format
99 * for(int i=0;i<NUM_DB_ROWS;i++){
100 * if(!nextLine(from) || !readVal(tempVals[i].blah)) readError(l,c,m);
101 *
102 * ....
103 *
104 * }
105 */
106
107 from.close();
108 return true;
109 #undef __METHOD__
110 }

```

4.3.2.11 bool tpcDewPointSender::readData (const char * *fileName*) [virtual]

Definition at line 116 of file tpcDewPointSender.cc.

```

116                                     {
117 #define __METHOD__ "readData(fileName)"
118
119 ifstream from(fileName);
120 if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
121
122 return readData(from); // user implemented file read
123 #undef __METHOD__
124 }

```

4.3.2.12 void tpcDewPointSender::readError (int *l*, char * *c*, char * *m*) [inline]

Definition at line 81 of file tpcDewPointSender.hh.

```

81                                     {
82 mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
83 }

```

4.3.2.13 bool tpcDewPointSender::readVal (long long & *value*)

Definition at line 269 of file tpcDewPointSender.cc.

```

269                                     {
270
271 if(!readAny())return false;

```

```
272 char* store[256];
273 value=strtoll(ptr2,store,10);
274 if(strlen(*store)>0) return false; // value is not a number
275
276 return true;
277 };
```

4.3.2.14 bool tpcDewPointSender::readVal (long & value)

Definition at line 258 of file tpcDewPointSender.cc.

```
258                                     {
259
260     if(!readAny())return false;
261
262     char* store[256];
263     value=strtol(ptr2,store,10);
264     if(strlen(*store)>0) return false; // value is not a number
265
266     return true;
267 };
```

4.3.2.15 bool tpcDewPointSender::readVal (int & value)

Definition at line 247 of file tpcDewPointSender.cc.

```
247                                     {
248
249     if(!readAny()) return false;
250
251     char* store[256];
252     value=(int)strtol(ptr2,store,10);
253     if(strlen(*store)>0) return false; // value is not a number
254
255     return true;
256 };
```

4.3.2.16 bool tpcDewPointSender::readVal (short & value)

Definition at line 236 of file tpcDewPointSender.cc.

```
236                                     {
237
238     if(!readAny()) return false;
239
240     char* store[256];
241     value=(short)strtol(ptr2,store,10);
242     if(strlen(*store)>0) return false; // value is not a number
243
244     return true;
245 };
```

4.3.2.17 bool tpcDewPointSender::readVal (double & *value*)

Definition at line 225 of file tpcDewPointSender.cc.

```
225         {
226
227     if(!readAny())return false;
228
229     char* store[256];
230     value=strtod(ptr2,store);
231     if(strlen(*store)>0) return false; // value is not a number
232
233     return true;
234 };
```

4.3.2.18 bool tpcDewPointSender::readVal (float & *value*)

Definition at line 214 of file tpcDewPointSender.cc.

```
214         {
215
216     if(!readAny()) return false;
217
218     char* store[256];
219     value=(float)strtod(ptr2,store);
220     if(strlen(*store)>0) return false; // value is not a number
221
222     return true;
223 };
```

4.3.2.19 bool tpcDewPointSender::readVal (char *& *value*)

Definition at line 207 of file tpcDewPointSender.cc.

```
207         {
208
209     if(!readAny()) return false;
210     strcpy(value,ptr2);
211     return true;
212 }
```

4.3.2.20 bool tpcDewPointSender::updateDb (const char * *fileName*) [virtual]

Definition at line 127 of file tpcDewPointSender.cc.

```
127         {
128     #define __METHOD__ "updateDb(filename)"
129
130     if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
131
132     int* elements;
133     tpcDewPoint* vals;
134     int numRows = 0;
135
136     if(writeRequired()){
137
```

```

138     numRows=NUM_DB_ROWS;
139     elements=elementList;
140     vals = tempVals;
141
142 } else {
143
144     for(int i=0; i<NUM_DB_ROWS; i++){
145         if(hasChanged(i)){
146             updateElements[numRows]=elementList[i];
147             updateVals[numRows] = tempVals[i];
148             previousVals[i]=tempVals[i];
149             numRows++;
150         }
151     }
152
153     elements = updateElements;
154     vals     = updateVals;
155 }
156
157 if(numRows==0) return sendMess(" No update required for",mbaseName,dbMDebug,__LINE__,__CLASS__,__MET
158
159 //char mess[256];
160 ostreamstream sn;
161 sn<<"Will Update "<<numRows<<" of "<<NUM_DB_ROWS<<" rows ";
162 sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
163
164 StDbTable* dbTable=node->findTable("tpcDewPoint");
165 dbTable->SetTable((char*)vals, numRows, elements);
166 mgr->setStoreTime(writeTime);
167
168 if(!mgr->storeDbTable(dbTable)) {
169     addBackLog(writeTime);
170     return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
171 }
172
173 if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
174
175 return true;
176 #undef __METHOD__
177 }

```

4.3.3 Member Data Documentation

4.3.3.1 int [tpcDewPointSender::elementList](#)[NUM_DB_ROWS] [protected]

Definition at line 24 of file tpcDewPointSender.hh.

4.3.3.2 float [tpcDewPointSender::hdriftLimit](#) [protected]

dito

Definition at line 36 of file tpcDewPointSender.hh.

4.3.3.3 char [tpcDewPointSender::mline](#)[256] [protected]

Definition at line 29 of file tpcDewPointSender.hh.

4.3.3.4 bool [tpcDewPointSender::mreadStatus](#) [protected]

Definition at line 28 of file tpcDewPointSender.hh.

4.3.3.5 tpcDewPoint [tpcDewPointSender::previousVals\[NUM_DB_ROWS\]](#) [protected]

Definition at line 22 of file tpcDewPointSender.hh.

4.3.3.6 char* [tpcDewPointSender::ptr1](#) [protected]

Definition at line 31 of file tpcDewPointSender.hh.

4.3.3.7 char * [tpcDewPointSender::ptr2](#) [protected]

Definition at line 31 of file tpcDewPointSender.hh.

4.3.3.8 float [tpcDewPointSender::tdriftLimit](#) [protected]

Definition at line 37 of file tpcDewPointSender.hh.

4.3.3.9 tpcDewPoint [tpcDewPointSender::tempVals\[NUM_DB_ROWS\]](#) [protected]

Definition at line 23 of file tpcDewPointSender.hh.

4.3.3.10 char [tpcDewPointSender::tmpline\[256\]](#) [protected]

Definition at line 30 of file tpcDewPointSender.hh.

4.3.3.11 int [tpcDewPointSender::updateElements\[NUM_DB_ROWS\]](#) [protected]

Definition at line 26 of file tpcDewPointSender.hh.

4.3.3.12 tpcDewPoint [tpcDewPointSender::updateVals\[NUM_DB_ROWS\]](#) [protected]

Definition at line 25 of file tpcDewPointSender.hh.

The documentation for this class was generated from the following files:

- [tpcDewPointSender.hh](#)
- [tpcDewPointSender.cc](#)
- [tpcDewPointSender_i.cc](#)

4.4 tpcFieldCageSender Class Reference

```
#include <tpcFieldCageSender.hh>
```

Public Member Functions

- [tpcFieldCageSender](#) (const char *localDir)
- virtual [~tpcFieldCageSender](#) ()
- virtual void [initTable](#) ()
- virtual void [initTags](#) ()
- virtual void [initDataBase](#) ()
- virtual bool [loadUserControls](#) (const char *name, const char *value)
- virtual void [initQuery](#) ()
- virtual bool [queryData](#) ()
- virtual bool [readData](#) (const char *fileName)
- virtual bool [updateDb](#) (const char *fileName)
- virtual bool [readData](#) (ifstream &from)
- virtual bool [hasChanged](#) (int rowNumber)
- char * [readAny](#) ()
- bool [readVal](#) (char *&value)
- bool [readVal](#) (float &value)
- bool [readVal](#) (double &value)
- bool [readVal](#) (short &value)
- bool [readVal](#) (int &value)
- bool [readVal](#) (long &value)
- bool [readVal](#) (long long &value)
- bool [nextLine](#) (ifstream &from)
- void [readError](#) (int l, char *c, char *m)

Protected Attributes

- tpcFieldCage [previousVals](#) [NUM_DB_ROWS]
- tpcFieldCage [tempVals](#) [NUM_DB_ROWS]
- int [elementList](#) [NUM_DB_ROWS]
- tpcFieldCage [updateVals](#) [NUM_DB_ROWS]
- int [updateElements](#) [NUM_DB_ROWS]
- bool [mreadStatus](#)
- char [mline](#) [256]
- char [tmpline](#) [256]
- char * [ptr1](#)
- char * [ptr2](#)
- float [vdriftLimit](#)
ditto
- float [cdriftLimit](#)
- float [gsdriftLimit](#)

4.4.1 Constructor & Destructor Documentation

4.4.1.1 tpcFieldCageSender::tpcFieldCageSender (const char * localDir)

Definition at line 19 of file tpcFieldCageSender.cc.

```

19                                     {
20
21     initTags();
22     if(localDir) cd(localDir);// note this ignores the sub dir tag
23     init("tpcFieldCage");// setup the file I/O
24     initDataBase();           // database connections
25     initTable();              // table definitions
26
27 }
```

4.4.1.2 virtual tpcFieldCageSender::~tpcFieldCageSender () [inline, virtual]

Definition at line 44 of file tpcFieldCageSender.hh.

```
44 {};
```

4.4.2 Member Function Documentation

4.4.2.1 bool tpcFieldCageSender::hasChanged (int rowNumber) [virtual]

Definition at line 144 of file tpcFieldCageSender_i.cc.

```

144                                     {
145
146     tpcFieldCage* pre=&previousVals[rowNumber];
147     tpcFieldCage* cur=&tempVals[rowNumber];
148
149     if(fabs(pre->voltageInnerFieldCageWest_1-cur->voltageInnerFieldCageWest_1)>=cdriftLimit)return true;
150     if(fabs(pre->voltageOuterFieldCageWest_1-cur->voltageOuterFieldCageWest_1)>=cdriftLimit)return true;
151     if(fabs(pre->voltageInnerFieldCageEast_1-cur->voltageInnerFieldCageEast_1)>=cdriftLimit)return true;
152     if(fabs(pre->voltageOuterFieldCageEast_1-cur->voltageOuterFieldCageEast_1)>=cdriftLimit)return true;
153     if(fabs(pre->voltageOuterFieldCageWest_0-cur->voltageOuterFieldCageWest_0)>=vdriftLimit)return true;
154     if(fabs(pre->voltageOuterFieldCageWest_1-cur->voltageOuterFieldCageWest_1)>=vdriftLimit)return true;
155     if(fabs(pre->voltageInnerFieldCageWest_0-cur->voltageInnerFieldCageWest_0)>=vdriftLimit)return true;
156     if(fabs(pre->voltageInnerFieldCageWest_1-cur->voltageInnerFieldCageWest_1)>=vdriftLimit)return true;
157     if(fabs(pre->voltageOuterFieldCageEast_0-cur->voltageOuterFieldCageEast_0)>=vdriftLimit)return true;
158     if(fabs(pre->voltageOuterFieldCageEast_1-cur->voltageOuterFieldCageEast_1)>=vdriftLimit)return true;
159     if(fabs(pre->voltageInnerFieldCageEast_0-cur->voltageInnerFieldCageEast_0)>=vdriftLimit)return true;
160     if(fabs(pre->voltageInnerFieldCageEast_1-cur->voltageInnerFieldCageEast_1)>=vdriftLimit)return true;
161     if(fabs(pre->groundShellCurrent-cur->groundShellCurrent)>=gscdriftLimit)return true;
162
163
164     /* example ... note -> change to any element requires db-update
165     * and thus returns true immediately
166     *
167     *if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
168     *if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
169     *
170     * ....
171     */
172
173     return false;
174 }
```

4.4.2.2 void tpcFieldCageSender::initDataBase () [virtual]

Definition at line 75 of file tpcFieldCageSender.cc.

```

75                                     {
76 #define __METHOD__ "initDataBase()"
77
78 /* More than an example... swap user & dbTrg as per subsystem*/
79 mgr->setUser("stardb","");
80 StDbType dbT = dbConditions;
81 StDbDomain dbD = dbTpc;
82
83 if( !( node = mgr->initConfig(dbT,dbD)) )
84     sendMess("Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal,__LINE__,__CLASS__,__METHOD__);
85
86 #undef __METHOD__
87 }
```

4.4.2.3 void tpcFieldCageSender::initQuery () [virtual]

Definition at line 53 of file tpcFieldCageSender_i.cc.

```

53                                     {
54 #define __METHOD__ "initQuery()"
55
56     ofstream to(queryFile);
57
58     if(!to.is_open()){
59         sendMess("Open Failed ",queryFile,dbMFatal,__LINE__,__CLASS__,__METHOD__);
60         return;
61     }
62
63     to << "TpcLvfcIW_OFC_1" << endl;
64     to << "TpcLvfcIW_IFC_1" << endl;
65     to << "TpcLvfcIE_OFC_1" << endl;
66     to << "TpcLvfcIE_IFC_1" << endl;
67     to << "TpcLvfcVW_OFC_0" << endl;
68     to << "TpcLvfcVW_OFC_1" << endl;
69     to << "TpcLvfcVW_IFC_0" << endl;
70     to << "TpcLvfcVW_IFC_1" << endl;
71     to << "TpcLvfcVE_OFC_0" << endl;
72     to << "TpcLvfcVE_OFC_1" << endl;
73     to << "TpcLvfcVE_IFC_0" << endl;
74     to << "TpcLvfcVE_IFC_1" << endl;
75     to << "TpcLvfcGV_I" << endl;
76     to << "CathodeVoltageReadback" << endl;
77     to << "CathodeCurrentReadback" << endl;
78     to << "CathodeVoltageSetpoint" << endl;
79     to << "CathodeCurrentLimit" << endl;
80     to << "TimeOfFlight" << endl;
81
82 /* example
83 *     for(int i=0;i<16;i++){
84 *         to<<"TRGhv:SUB_RD_V_1:"<<i<<".E"<<endl;
85 *         to<<"TRGhv:SUB_RD_V_1:"<<i<<".F"<<endl;
86 *         ....
87 *     }
88 * */
89 */
90
91     to.close();
92
93 #undef __METHOD__
94 }
```

4.4.2.4 void tpcFieldCageSender::initTable () [virtual]

Definition at line 30 of file tpcFieldCageSender.cc.

```

30                                     {
31 #define __METHOD__ "initTable()"
32
33     StDbTable* table=0;
34     if(!(table=node->addDbTable("tpcFieldCage")))
35         sendMess("Could not find table=tpcFieldCage",dbMFatal,__LINE__,__CLASS__,__METHOD__);
36
37     memset(tempVals,0,NUM_DB_ROWS*sizeof(tpcFieldCage));
38     memset(previousVals,0,NUM_DB_ROWS*sizeof(tpcFieldCage));
39
40     int nrows;
41     int* elist = table->getElementID(nrows);
42     if(nrows!=NUM_DB_ROWS){
43         //char mess[256];
44         ostream ms;
45         ms<<"Db rows("<<nrows<<") != compiled("<<NUM_DB_ROWS<<")";
46         sendMess((ms.str()).c_str(),dbMFatal,__LINE__,__CLASS__,__METHOD__);
47     }
48     memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
49
50     unsigned int timestamp=time(NULL);
51     mgr->setRequestTime(timestamp);
52     if(mgr->fetchDbTable(table)){
53         tpcFieldCage* thv = (tpcFieldCage*)table->GetTable();
54         memcpy(previousVals,thv,nrows*sizeof(tpcFieldCage));
55     }
56
57 #undef __METHOD__
58 };

```

4.4.2.5 void tpcFieldCageSender::initTags () [virtual]

Definition at line 66 of file tpcFieldCageSender.cc.

```

66                                     {
67 /* more than an example -> swap "trg" to your subsys & add to email list*/
68     setEmailTo("porter@bnl.gov");
69     setDomainName("tpc");
70
71 }

```

4.4.2.6 bool tpcFieldCageSender::loadUserControls (const char * name, const char * value) [virtual]

Definition at line 20 of file tpcFieldCageSender_i.cc.

```

20                                     {
21 #define __METHOD__ "loadUserControls(name,value)"
22
23 /* more than an example ... swap driftLimit to yours
24 * and duplicate this structure for each selection criteria
25 if(strstr(name,"driftLimit")){
26     driftLimit=atof(value);
27     sendMess("driftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
28     return true;

```

```

29  }
30  */
31  if(strstr(name,"vdriftLimit")){
32      vdriftLimit=atof(value);
33      sendMess("vdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
34      return true;
35  }
36  if(strstr(name,"cdriftLimit")){
37      cdriftLimit=atof(value);
38      sendMess("cdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
39      return true;
40  }
41  if(strstr(name,"gscdriftLimit")){
42      gscdriftLimit=atof(value);
43      sendMess("gscdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
44      return true;
45  }
46
47  return false;
48  #undef __METHOD__
49  }

```

4.4.2.7 bool tpcFieldCageSender::nextLine (ifstream & from) [inline]

Definition at line 77 of file tpcFieldCageSender.hh.

```

77                                     {
78  if(!from.getline(mline,255))return false;
79  return true;
80  }

```

4.4.2.8 bool tpcFieldCageSender::queryData () [virtual]

Definition at line 91 of file tpcFieldCageSender.cc.

```

91                                     {
92  #define __METHOD__ "queryData()"
93
94  /*
95   * MORE THAN AN EXAMPLE...
96   * IF Standard SC-Query via "caGet" then,
97   * no need to change this method AT ALL
98   *
99   */
100
101  writeTime = (unsigned int)time(NULL); //for database write time
102
103  //char systemCmd[1024];
104  ostringstream scmd;
105  scmd<<"caGet " <<queryFile<<" " <<dataFile;
106
107  if(system((scmd.str()).c_str()))
108      return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
109
110  return true;
111  #undef __METHOD__
112  };

```

4.4.2.9 char * tpcFieldCageSender::readAny ()

Definition at line 197 of file tpcFieldCageSender.cc.

```

197         {
198
199     strcpy(tmpLine,mLine);
200     ptr1=tmpLine;
201     ptr2=strtok(ptr1," ");
202     if(!ptr2) return ptr2;
203     ptr2=strtok(NULL," ");
204     return ptr2;
205 }

```

4.4.2.10 bool tpcFieldCageSender::readData (ifstream & from) [virtual]

Definition at line 99 of file tpcFieldCageSender_i.cc.

```

99         {
100 #define __METHOD__ "readData(ifstream)"
101
102 mreadStatus=true;
103 memset(tempVals,0,NUM_DB_ROWS*sizeof(tpcFieldCage));
104
105
106 int i=0;
107 char* c=__CLASS__;
108 char* m=__METHOD__;
109
110     if(!nextLine(from) || !readVal(tempVals[0].currentInnerFieldCageWest_1)) readError(__LINE__,c,m);
111     if(!nextLine(from) || !readVal(tempVals[0].currentOuterFieldCageWest_1)) readError(__LINE__,c,m);
112     if(!nextLine(from) || !readVal(tempVals[0].currentInnerFieldCageEast_1)) readError(__LINE__,c,m);
113     if(!nextLine(from) || !readVal(tempVals[0].currentOuterFieldCageEast_1)) readError(__LINE__,c,m);
114     if(!nextLine(from) || !readVal(tempVals[0].voltageOuterFieldCageWest_0)) readError(__LINE__,c,m);
115     if(!nextLine(from) || !readVal(tempVals[0].voltageOuterFieldCageWest_1)) readError(__LINE__,c,m);
116     if(!nextLine(from) || !readVal(tempVals[0].voltageInnerFieldCageWest_0)) readError(__LINE__,c,m);
117     if(!nextLine(from) || !readVal(tempVals[0].voltageInnerFieldCageWest_1)) readError(__LINE__,c,m);
118     if(!nextLine(from) || !readVal(tempVals[0].voltageOuterFieldCageEast_0)) readError(__LINE__,c,m);
119     if(!nextLine(from) || !readVal(tempVals[0].voltageOuterFieldCageEast_1)) readError(__LINE__,c,m);
120     if(!nextLine(from) || !readVal(tempVals[0].voltageInnerFieldCageEast_0)) readError(__LINE__,c,m);
121     if(!nextLine(from) || !readVal(tempVals[0].voltageInnerFieldCageEast_1)) readError(__LINE__,c,m);
122     if(!nextLine(from) || !readVal(tempVals[0].groundShellCurrent)) readError(__LINE__,c,m);
123     if(!nextLine(from) || !readVal(tempVals[0].CathodeVoltageReadback)) readError(__LINE__,c,m);
124     if(!nextLine(from) || !readVal(tempVals[0].CathodeVoltageSetpoint)) readError(__LINE__,c,m);
125     if(!nextLine(from) || !readVal(tempVals[0].CathodeCurrentLimit)) readError(__LINE__,c,m);
126     if(!nextLine(from) || !readVal(tempVals[0].TimeOfFlight)) readError(__LINE__,c,m);
127
128 /* example format
129 * for(int i=0;i<NUM_DB_ROWS;i++){
130 * if(!nextLine(from) || !readVal(tempVals[i].blah)) readError(l,c,m);
131 *
132 *     ....
133 *
134 * }
135 */
136
137     from.close();
138     return true;
139 #undef __METHOD__
140 }

```

4.4.2.11 bool tpcFieldCageSender::readData (const char **fileName*) [virtual]

Definition at line 116 of file tpcFieldCageSender.cc.

```

116                                     {
117 #define __METHOD__ "readData(fileName)"
118
119     ifstream from(fileName);
120     if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
121
122     return readData(from); // user implemented file read
123 #undef __METHOD__
124 }
```

4.4.2.12 void tpcFieldCageSender::readError (int *l*, char **c*, char **m*) [inline]

Definition at line 82 of file tpcFieldCageSender.hh.

```

82                                     {
83     mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
84 }
```

4.4.2.13 bool tpcFieldCageSender::readVal (long long & *value*)

Definition at line 269 of file tpcFieldCageSender.cc.

```

269                                     {
270
271     if(!readAny())return false;
272     char* store[256];
273     value=strtoll(ptr2,store,10);
274     if(strlen(*store)>0) return false; // value is not a number
275
276     return true;
277 };
```

4.4.2.14 bool tpcFieldCageSender::readVal (long & *value*)

Definition at line 258 of file tpcFieldCageSender.cc.

```

258                                     {
259
260     if(!readAny())return false;
261
262     char* store[256];
263     value=strtol(ptr2,store,10);
264     if(strlen(*store)>0) return false; // value is not a number
265
266     return true;
267 };
```

4.4.2.15 bool tpcFieldCageSender::readVal (int & value)

Definition at line 247 of file tpcFieldCageSender.cc.

```
247                                     {
248
249     if(!readAny()) return false;
250
251     char* store[256];
252     value=(int)strtol(ptr2,store,10);
253     if(strlen(*store)>0) return false; // value is not a number
254
255     return true;
256 };
```

4.4.2.16 bool tpcFieldCageSender::readVal (short & value)

Definition at line 236 of file tpcFieldCageSender.cc.

```
236                                     {
237
238     if(!readAny()) return false;
239
240     char* store[256];
241     value=(short)strtol(ptr2,store,10);
242     if(strlen(*store)>0) return false; // value is not a number
243
244     return true;
245 };
```

4.4.2.17 bool tpcFieldCageSender::readVal (double & value)

Definition at line 225 of file tpcFieldCageSender.cc.

```
225                                     {
226
227     if(!readAny())return false;
228
229     char* store[256];
230     value=strtod(ptr2,store);
231     if(strlen(*store)>0) return false; // value is not a number
232
233     return true;
234 };
```

4.4.2.18 bool tpcFieldCageSender::readVal (float & value)

Definition at line 214 of file tpcFieldCageSender.cc.

```
214                                     {
215
216     if(!readAny()) return false;
217
218     char* store[256];
219     value=(float)strtod(ptr2,store);
220     if(strlen(*store)>0) return false; // value is not a number
221
222     return true;
223 };
```

4.4.2.19 bool tpcFieldCageSender::readVal (char *& value)

Definition at line 207 of file tpcFieldCageSender.cc.

```

207                                     {
208
209     if(!readAny()) return false;
210     strcpy(value,ptr2);
211     return true;
212 }
```

4.4.2.20 bool tpcFieldCageSender::updateDb (const char *fileName) [virtual]

Definition at line 127 of file tpcFieldCageSender.cc.

```

127                                     {
128 #define __METHOD__ "updateDb(filename)"
129
130     if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
131
132     int* elements;
133     tpcFieldCage* vals;
134     int numRows = 0;
135
136     if(writeRequired()){
137
138         numRows=NUM_DB_ROWS;
139         elements=elementList;
140         vals = tempVals;
141
142     } else {
143
144         for(int i=0; i<NUM_DB_ROWS; i++){
145             if(hasChanged(i)){
146                 updateElements[numRows]=elementList[i];
147                 updateVals[numRows] = tempVals[i];
148                 previousVals[i]=tempVals[i];
149                 numRows++;
150             }
151         }
152
153         elements = updateElements;
154         vals      = updateVals;
155     }
156
157     if(numRows==0) return sendMess(" No update required for",mbaseName,dbMDebug,__LINE__,__CLASS__,__METHOD__);
158
159     //char mess[256];
160     ostringstream sn;
161     sn<<"Will Update "<<numRows<<" of "<<NUM_DB_ROWS<<" rows ";
162     sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
163
164     StDbTable* dbTable=node->findTable("tpcFieldCage");
165     dbTable->SetTable((char*)vals, numRows, elements);
166     mgr->setStoreTime(writeTime);
167
168     if(!mgr->storeDbTable(dbTable)) {
169         addBackLog(writeTime);
170         return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
171     }
172
173     if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
174 }
```

```
175     return true;
176     #undef __METHOD__
177 }
```

4.4.3 Member Data Documentation

4.4.3.1 float [tpcFieldCageSender::cdriftLimit](#) [protected]

Definition at line 38 of file tpcFieldCageSender.hh.

4.4.3.2 int [tpcFieldCageSender::elementList](#)[NUM_DB_ROWS] [protected]

Definition at line 24 of file tpcFieldCageSender.hh.

4.4.3.3 float [tpcFieldCageSender::gsdriftLimit](#) [protected]

Definition at line 39 of file tpcFieldCageSender.hh.

4.4.3.4 char [tpcFieldCageSender::mline](#)[256] [protected]

Definition at line 29 of file tpcFieldCageSender.hh.

4.4.3.5 bool [tpcFieldCageSender::mreadStatus](#) [protected]

Definition at line 28 of file tpcFieldCageSender.hh.

4.4.3.6 tpcFieldCage [tpcFieldCageSender::previousVals](#)[NUM_DB_ROWS] [protected]

Definition at line 22 of file tpcFieldCageSender.hh.

4.4.3.7 char* [tpcFieldCageSender::ptr1](#) [protected]

Definition at line 31 of file tpcFieldCageSender.hh.

4.4.3.8 char * [tpcFieldCageSender::ptr2](#) [protected]

Definition at line 31 of file tpcFieldCageSender.hh.

4.4.3.9 tpcFieldCage [tpcFieldCageSender::tempVals](#)[NUM_DB_ROWS] [protected]

Definition at line 23 of file tpcFieldCageSender.hh.

4.4.3.10 char [tpcFieldCageSender::tmpline](#)[256] [protected]

Definition at line 30 of file tpcFieldCageSender.hh.

4.4.3.11 int [tpcFieldCageSender::updateElements](#)[NUM_DB_ROWS] [protected]

Definition at line 26 of file tpcFieldCageSender.hh.

4.4.3.12 tpcFieldCage [tpcFieldCageSender::updateVals](#)[NUM_DB_ROWS] [protected]

Definition at line 25 of file tpcFieldCageSender.hh.

4.4.3.13 float [tpcFieldCageSender::vdriftLimit](#) [protected]

dito

Definition at line 37 of file tpcFieldCageSender.hh.

The documentation for this class was generated from the following files:

- [tpcFieldCageSender.hh](#)
- [tpcFieldCageSender.cc](#)
- [tpcFieldCageSender_i.cc](#)

4.5 tpcGainMonitorSender Class Reference

```
#include <tpcGainMonitorSender.hh>
```

Public Member Functions

- [tpcGainMonitorSender](#) (const char *localDir)
- virtual [~tpcGainMonitorSender](#) ()
- virtual void [initTable](#) ()
- virtual void [initTags](#) ()
- virtual void [initDataBase](#) ()
- virtual bool [loadUserControls](#) (const char *name, const char *value)
- virtual void [initQuery](#) ()
- virtual bool [queryData](#) ()
- virtual bool [readData](#) (const char *fileName)
- virtual bool [updateDb](#) (const char *fileName)
- virtual bool [readData](#) (ifstream &from)
- virtual bool [hasChanged](#) (int rowNumber)
- char * [readAny](#) ()
- bool [readVal](#) (char *&value)
- bool [readVal](#) (float &value)
- bool [readVal](#) (double &value)
- bool [readVal](#) (short &value)
- bool [readVal](#) (int &value)
- bool [readVal](#) (long &value)
- bool [readVal](#) (long long &value)
- bool [nextLine](#) (ifstream &from)
- void [readError](#) (int l, char *c, char *m)

Protected Attributes

- tpcGainMonitor [previousVals](#) [NUM_DB_ROWS]
- tpcGainMonitor [tempVals](#) [NUM_DB_ROWS]
- int [elementList](#) [NUM_DB_ROWS]
- tpcGainMonitor [updateVals](#) [NUM_DB_ROWS]
- int [updateElements](#) [NUM_DB_ROWS]
- bool [mreadStatus](#)
- char [mline](#) [256]
- char [tmpline](#) [256]
- char * [ptr1](#)
- char * [ptr2](#)
- float [widthLimit](#)

ditto

4.5.1 Constructor & Destructor Documentation

4.5.1.1 tpcGainMonitorSender::tpcGainMonitorSender (const char * localDir)

Definition at line 19 of file tpcGainMonitorSender.cc.

```

19                                     {
20
21     initTags();
22     if(localDir) cd(localDir);// note this ignores the sub dir tag
23     init("tpcGainMonitor"); // setup the file I/O
24     initDataBase();         // database connections
25     initTable();           // table definitions
26
27 }
```

4.5.1.2 virtual tpcGainMonitorSender::~~tpcGainMonitorSender () [inline, virtual]

Definition at line 41 of file tpcGainMonitorSender.hh.

```
41 {};
```

4.5.2 Member Function Documentation

4.5.2.1 bool tpcGainMonitorSender::hasChanged (int rowNumber) [virtual]

Definition at line 105 of file tpcGainMonitorSender_i.cc.

```

105                                     {
106
107     tpcGainMonitor* pre=&previousVals[rowNumber];
108     tpcGainMonitor* cur=&tempVals[rowNumber];
109
110     if(fabs(pre->width-cur->width)>=widthLimit) return true;
111     /* example ... note -> change to any element requires db-update
112     * and thus returns true immediately
113     *
114     *if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
115     *if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
116     *
117     * ....
118     */
119
120     return false;
121 }
```

4.5.2.2 void tpcGainMonitorSender::initDataBase () [virtual]

Definition at line 75 of file tpcGainMonitorSender.cc.

```

75                                     {
76     #define __METHOD__ "initDataBase()"
77
78     /* More than an example... swap user & dbTrg as per subsystem*/
79     mgr->setUser("stardb","");
```

```

80  StDbType   dbT = dbConditions;
81  StDbDomain dbD = dbTpc;
82
83  if( !( node = mgr->initConfig(dbT,dbD) ) )
84      sendMess("Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal, __LINE__, __CLASS__, __METHOD__);
85
86 #undef __METHOD__
87 }

```

4.5.2.3 void tpcGainMonitorSender::initQuery () [virtual]

Definition at line 43 of file tpcGainMonitorSender_i.cc.

```

43                                     {
44 #define __METHOD__ "initQuery()"
45
46     ofstream to(queryFile);
47
48     if(!to.is_open()){
49         sendMess("Open Failed ",queryFile,dbMFatal, __LINE__, __CLASS__, __METHOD__);
50         return;
51     }
52
53     to<<"gain:peak_center"<<endl;
54     to<<"gain:peak_height"<<endl;
55     to<<"gain:peak_width"<<endl;
56
57 /* example
58 *     for(int i=0;i<16;i++){
59 *         to<<"TRGhv:SUB_RD_V_1:"<<i<<".E"<<endl;
60 *         to<<"TRGhv:SUB_RD_V_1:"<<i<<".F"<<endl;
61 *         ....
62 *     }
63 *
64 */
65
66     to.close();
67
68 #undef __METHOD__
69 }

```

4.5.2.4 void tpcGainMonitorSender::initTable () [virtual]

Definition at line 30 of file tpcGainMonitorSender.cc.

```

30                                     {
31 #define __METHOD__ "initTable()"
32
33     StDbTable* table=0;
34     if(!(table=node->addDbTable("tpcGainMonitor")))
35         sendMess("Could not find table=tpcGainMonitor",dbMFatal, __LINE__, __CLASS__, __METHOD__);
36
37     memset(tempVals,0,NUM_DB_ROWS*sizeof(tpcGainMonitor));
38     memset(previousVals,0,NUM_DB_ROWS*sizeof(tpcGainMonitor));
39
40     int nrows;
41     int* elist = table->getElementID(nrows);
42     if(nrows!=NUM_DB_ROWS){
43         //char mess[256];
44         ostringstream ms;
45         ms<<"Db rows("<<nrows<<") != compiled("<<NUM_DB_ROWS<<")";

```

```

46     sendMess((ms.str()).c_str(),dbMFatal,__LINE__,__CLASS__,__METHOD__);
47 }
48 memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
49
50 unsigned int timestamp=time(NULL);
51 mgr->setRequestTime(timestamp);
52 if(mgr->fetchDbTable(table)){
53     tpcGainMonitor* thv = (tpcGainMonitor*)table->GetTable();
54     memcpy(previousVals,thv,nrows*sizeof(tpcGainMonitor));
55 }
56
57 #undef __METHOD__
58 };

```

4.5.2.5 void tpcGainMonitorSender::initTags () [virtual]

Definition at line 66 of file tpcGainMonitorSender.cc.

```

66                                     {
67     /* more than an example -> swap "trg" to your subsys & add to email list*/
68     setEmailTo("porter@bnl.gov");
69     setDomainName("tpc");
70
71 }

```

4.5.2.6 bool tpcGainMonitorSender::loadUserControls (const char * name, const char * value) [virtual]

Definition at line 20 of file tpcGainMonitorSender_i.cc.

```

20                                     {
21 #define __METHOD__ "loadUserControls(name,value)"
22
23 /* more than an example ... swap driftLimit to yours
24 * and duplicate this structure for each selection criteria
25 if(strstr(name,"driftLimit")){
26     driftLimit=atof(value);
27     sendMess("driftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
28     return true;
29 }
30 */
31 if(strstr(name,"widthLimit")){
32     widthLimit=atof(value);
33     sendMess("widthLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
34     return true;
35 }
36
37 return false;
38 #undef __METHOD__
39 }

```

4.5.2.7 bool tpcGainMonitorSender::nextLine (ifstream & from) [inline]

Definition at line 74 of file tpcGainMonitorSender.hh.

```

74                                     {
75     if(!from.getline(mline,255))return false;
76     return true;
77 }

```

4.5.2.8 bool tpcGainMonitorSender::queryData () [virtual]

Definition at line 91 of file tpcGainMonitorSender.cc.

```

91                                     {
92 #define __METHOD__ "queryData()"
93
94 /*
95  * MORE THAN AN EXAMPLE...
96  * IF Standard SC-Query via "caGet" then,
97  * no need to change this method AT ALL
98  *
99  */
100
101 writeTime = (unsigned int)time(NULL);      //for database write time
102
103 //char systemCmd[1024];
104 ostringstream scmd;
105 scmd<<"caGet "<<queryFile<<" "<<dataFile;
106
107 if(system((scmd.str()).c_str()))
108     return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
109
110 return true;
111 #undef __METHOD__
112 };

```

4.5.2.9 char * tpcGainMonitorSender::readAny ()

Definition at line 197 of file tpcGainMonitorSender.cc.

```

197                                     {
198
199 strcpy(tmpline,mline);
200 ptr1=tmpline;
201 ptr2=strtok(ptr1," ");
202 if(!ptr2) return ptr2;
203 ptr2=strtok(NULL," ");
204 return ptr2;
205 }

```

4.5.2.10 bool tpcGainMonitorSender::readData (ifstream &from) [virtual]

Definition at line 74 of file tpcGainMonitorSender_i.cc.

```

74                                     {
75 #define __METHOD__ "readData(ifstream)"
76
77 mreadStatus=true;
78 memset(tempVals,0,NUM_DB_ROWS*sizeof(tpcGainMonitor));
79
80
81
82 int i=0;
83 char* c=__CLASS__;
84 char* m=__METHOD__;
85
86 if(!nextLine(from) || !readVal(tempVals[0].center))readError(__LINE__,c,m);
87 if(!nextLine(from) || !readVal(tempVals[0].height))readError(__LINE__,c,m);

```

```

88     if(!nextLine(from) || !readVal(tempVals[0].width))readError(__LINE__,c,m);
89 /* example format
90 *   for(int i=0;i<NUM_DB_ROWS;i++){
91 *   if(!nextLine(from) || !readVal(tempVals[i].blah)) readError(l,c,m);
92 *
93 *     ....
94 *
95 *   }
96 */
97
98   from.close();
99   return true;
100 #undef __METHOD__
101 }

```

4.5.2.11 bool tpcGainMonitorSender::readData (const char * *fileName*) [virtual]

Definition at line 116 of file tpcGainMonitorSender.cc.

```

116                                     {
117 #define __METHOD__ "readData(fileName)"
118
119   ifstream from(fileName);
120   if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
121
122   return readData(from); // user implemented file read
123 #undef __METHOD__
124 }

```

4.5.2.12 void tpcGainMonitorSender::readError (int *l*, char * *c*, char * *m*) [inline]

Definition at line 79 of file tpcGainMonitorSender.hh.

```

79                                     {
80   mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
81 }

```

4.5.2.13 bool tpcGainMonitorSender::readVal (long long & *value*)

Definition at line 269 of file tpcGainMonitorSender.cc.

```

269                                     {
270
271   if(!readAny())return false;
272   char* store[256];
273   value=strtoll(ptr2,store,10);
274   if(strlen(*store)>0) return false; // value is not a number
275
276   return true;
277 };

```

4.5.2.14 bool tpcGainMonitorSender::readVal (long & *value*)

Definition at line 258 of file tpcGainMonitorSender.cc.

```
258         {
259
260     if(!readAny())return false;
261
262     char* store[256];
263     value=strtol(ptr2,store,10);
264     if(strlen(*store)>0) return false; // value is not a number
265
266     return true;
267 };
```

4.5.2.15 bool tpcGainMonitorSender::readVal (int & value)

Definition at line 247 of file tpcGainMonitorSender.cc.

```
247         {
248
249     if(!readAny()) return false;
250
251     char* store[256];
252     value=(int)strtol(ptr2,store,10);
253     if(strlen(*store)>0) return false; // value is not a number
254
255     return true;
256 };
```

4.5.2.16 bool tpcGainMonitorSender::readVal (short & value)

Definition at line 236 of file tpcGainMonitorSender.cc.

```
236         {
237
238     if(!readAny()) return false;
239
240     char* store[256];
241     value=(short)strtol(ptr2,store,10);
242     if(strlen(*store)>0) return false; // value is not a number
243
244     return true;
245 };
```

4.5.2.17 bool tpcGainMonitorSender::readVal (double & value)

Definition at line 225 of file tpcGainMonitorSender.cc.

```
225         {
226
227     if(!readAny())return false;
228
229     char* store[256];
230     value=strtod(ptr2,store);
231     if(strlen(*store)>0) return false; // value is not a number
232
233     return true;
234 };
```

4.5.2.18 bool tpcGainMonitorSender::readVal (float & value)

Definition at line 214 of file tpcGainMonitorSender.cc.

```

214         {
215
216     if(!readAny()) return false;
217
218     char* store[256];
219     value=(float)strtod(ptr2,store);
220     if(strlen(*store)>0) return false; // value is not a number
221
222     return true;
223 };

```

4.5.2.19 bool tpcGainMonitorSender::readVal (char *& value)

Definition at line 207 of file tpcGainMonitorSender.cc.

```

207         {
208
209     if(!readAny()) return false;
210     strcpy(value,ptr2);
211     return true;
212 }

```

4.5.2.20 bool tpcGainMonitorSender::updateDb (const char *fileName) [virtual]

Definition at line 127 of file tpcGainMonitorSender.cc.

```

127         {
128 #define __METHOD__ "updateDb(filename)"
129
130     if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
131
132     int* elements;
133     tpcGainMonitor* vals;
134     int numRows = 0;
135
136     if(writeRequired()){
137
138         numRows=NUM_DB_ROWS;
139         elements=elementList;
140         vals = tempVals;
141
142     } else {
143
144         for(int i=0; i<NUM_DB_ROWS; i++){
145             if(hasChanged(i)){
146                 updateElements[numRows]=elementList[i];
147                 updateVals[numRows] = tempVals[i];
148                 previousVals[i]=tempVals[i];
149                 numRows++;
150             }
151         }
152
153         elements = updateElements;
154         vals     = updateVals;
155     }

```

```

156
157     if(numRows==0) return sendMess(" No update required for",mbaseName,dbMDebug,__LINE__,__CLASS__,__MET
158
159     //char mess[256];
160     ostringstream sn;
161     sn<<"Will Update "<<numRows<<" of "<<NUM_DB_ROWS<<" rows ";
162     sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
163
164     StDbTable* dbTable=node->findTable("tpcGainMonitor");
165     dbTable->SetTable((char*)vals, numRows, elements);
166     mgr->setStoreTime(writeTime);
167
168     if(!mgr->storeDbTable(dbTable)) {
169         addBackLog(writeTime);
170         return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
171     }
172
173     if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
174
175     return true;
176 #undef __METHOD__
177 }

```

4.5.3 Member Data Documentation

4.5.3.1 int [tpcGainMonitorSender::elementList](#)[NUM_DB_ROWS] [protected]

Definition at line 24 of file tpcGainMonitorSender.hh.

4.5.3.2 char [tpcGainMonitorSender::mline](#)[256] [protected]

Definition at line 29 of file tpcGainMonitorSender.hh.

4.5.3.3 bool [tpcGainMonitorSender::mreadStatus](#) [protected]

Definition at line 28 of file tpcGainMonitorSender.hh.

4.5.3.4 tpcGainMonitor [tpcGainMonitorSender::previousVals](#)[NUM_DB_ROWS] [protected]

Definition at line 22 of file tpcGainMonitorSender.hh.

4.5.3.5 char* [tpcGainMonitorSender::ptr1](#) [protected]

Definition at line 31 of file tpcGainMonitorSender.hh.

4.5.3.6 char * [tpcGainMonitorSender::ptr2](#) [protected]

Definition at line 31 of file tpcGainMonitorSender.hh.

4.5.3.7 tpcGainMonitor [tpcGainMonitorSender::tempVals](#)[NUM_DB_ROWS] [protected]

Definition at line 23 of file tpcGainMonitorSender.hh.

4.5.3.8 char [tpcGainMonitorSender::tmpline\[256\]](#) [protected]

Definition at line 30 of file tpcGainMonitorSender.hh.

4.5.3.9 int [tpcGainMonitorSender::updateElements\[NUM_DB_ROWS\]](#) [protected]

Definition at line 26 of file tpcGainMonitorSender.hh.

4.5.3.10 [tpcGainMonitor](#) [tpcGainMonitorSender::updateVals\[NUM_DB_ROWS\]](#)
[protected]

Definition at line 25 of file tpcGainMonitorSender.hh.

4.5.3.11 float [tpcGainMonitorSender::widthLimit](#) [protected]

dito

Definition at line 35 of file tpcGainMonitorSender.hh.

The documentation for this class was generated from the following files:

- [tpcGainMonitorSender.hh](#)
- [tpcGainMonitorSender.cc](#)
- [tpcGainMonitorSender_i.cc](#)

4.6 tpcGasSender Class Reference

```
#include <tpcGasSender.h>
```

Public Member Functions

- [tpcGasSender](#) (const char *localDir)
ditto
- virtual [~tpcGasSender](#) ()
- virtual void [initTable](#) ()
- virtual void [initTags](#) ()
- virtual void [initDataBase](#) ()
- virtual bool [loadUserControls](#) (const char *name, const char *value)
- virtual void [initQuery](#) ()
- virtual bool [queryData](#) ()
- virtual bool [readData](#) (const char *fileName)
- virtual bool [updateDb](#) (const char *fileName)
- virtual bool [readData](#) (ifstream &from)
- virtual bool [hasChanged](#) (int rowNumber)
- char * [readAny](#) ()
- bool [readVal](#) (char *&value)
- bool [readVal](#) (float &value)
- bool [readVal](#) (double &value)
- bool [readVal](#) (short &value)
- bool [readVal](#) (int &value)
- bool [readVal](#) (long &value)
- bool [readVal](#) (long long &value)
- bool [nextLine](#) (ifstream &from)
- void [readError](#) (int l, char *c, char *m)

Protected Attributes

- tpcGas [previousVals](#) [NUM_DB_ROWS]
- tpcGas [tempVals](#) [NUM_DB_ROWS]
- int [elementList](#) [NUM_DB_ROWS]
- tpcGas [updateVals](#) [NUM_DB_ROWS]
- int [updateElements](#) [NUM_DB_ROWS]
- bool [mreadStatus](#)
- char [mline](#) [256]
- char [tmpline](#) [256]
- char * [ptr1](#)
- char * [ptr2](#)

4.6.1 Constructor & Destructor Documentation

4.6.1.1 tpcGasSender::tpcGasSender (const char * localDir)

dito

Definition at line 18 of file tpcGasSender.cc.

```

18         {
19
20     initTags();
21     if(localDir) cd(localDir); // note this ignores the sub dir tag
22     init("tpcGas"); // setup the file I/O
23     initDataBase(); // database connections
24     initTable(); // table definitions
25
26 }
```

4.6.1.2 virtual tpcGasSender::~tpcGasSender () [inline, virtual]

Definition at line 41 of file tpcGasSender.hh.

```
41 {};
```

4.6.2 Member Function Documentation

4.6.2.1 bool tpcGasSender::hasChanged (int rowNumber) [virtual]

Definition at line 193 of file tpcGasSender_i.cc.

```

193         {
194
195
196     return true;
197 }
```

4.6.2.2 void tpcGasSender::initDataBase () [virtual]

Definition at line 74 of file tpcGasSender.cc.

```

74         {
75     #define __METHOD__ "initDataBase()"
76
77     /* More than an example... swap user & dbTrg as per subsystem*/
78     mgr->setUser("stardb","");
79     StDbType dbT = dbConditions;
80     StDbDomain dbD = dbTpc;
81
82     if( !( node = mgr->initConfig(dbT,dbD) ) )
83         sendMess("Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal,__LINE__,__CLASS__,__METHOD__);
84
85     #undef __METHOD__
86 }
```

4.6.2.3 void tpcGasSender::initQuery () [virtual]

Definition at line 38 of file tpcGasSender_i.cc.

```
38         {
39 #define __METHOD__ "initQuery()"
40
41     ofstream to(queryFile);
42
43     if(!to.is_open()){
44         sendMess("Open Failed ",queryFile,dbMFatal,__LINE__,__CLASS__,__METHOD__);
45         return;
46     }
47
48
49     to << "cu_tpc_gas_CH4-P1" << endl;
50     to << "cu_tpc_gas_CH4-P2" << endl;
51     to << "cu_tpc_gas_O2-M1" << endl;
52     to << "cu_tpc_gas_O2-M5" << endl;
53     to << "cu_tpc_gas_CH4-M3" << endl;
54     to << "cu_tpc_gas_CH4-M4" << endl;
55     to << "cu_tpc_gas_PT-1" << endl;
56     to << "cu_tpc_gas_PT-2" << endl;
57     to << "cu_tpc_gas_PT-3" << endl;
58     // to << "cu_tpc_gas_PT-4" << endl;
59     to << "cu_tpc_gas_PT-5" << endl;
60     to << "cu_tpc_gas_PT-7" << endl;
61     to << "cu_tpc_gas_PT-8" << endl;
62     to << "cu_tpc_gas_PT-B" << endl;
63     // to << "cu_tpc_gas_PT-B2" << endl;
64     to << "cu_tpc_gas_PT-10" << endl;
65     to << "cu_tpc_gas_PT-9" << endl;
66     to << "cu_tpc_gas_PT-11" << endl;
67     to << "cu_tpc_gas_PI-8" << endl;
68     to << "cu_tpc_gas_PI-9" << endl;
69     to << "cu_tpc_gas_PI-10" << endl;
70     to << "cu_tpc_gas_PI-13" << endl;
71     to << "cu_tpc_gas_PI-14" << endl;
72     to << "cu_tpc_gas_PI-15" << endl;
73     to << "cu_tpc_gas_T1" << endl;
74     to << "cu_tpc_gas_T2" << endl;
75     to << "cu_tpc_gas_T3" << endl;
76     to << "cu_tpc_gas_T4" << endl;
77     to << "cu_tpc_gas_T5" << endl;
78     to << "cu_tpc_gas_T6" << endl;
79     to << "cu_tpc_gas_T7" << endl;
80     to << "cu_tpc_gas_T8" << endl;
81     to << "cu_tpc_gas_T9" << endl;
82     to << "cu_tpc_gas_T10" << endl;
83     to << "cu_tpc_gas_T11" << endl;
84     to << "cu_tpc_gas_T12" << endl;
85     to << "cu_tpc_gas_T13" << endl;
86     to << "cu_tpc_gas_T14" << endl;
87     to << "cu_tpc_gas_T15" << endl;
88     to << "cu_tpc_gas_T16" << endl;
89     to << "cu_tpc_gas_H2O-M2" << endl;
90     // to << "cu_tpc_gas_C2H6-M3" << endl;
91     to << "cu_tpc_gas_FM-1" << endl;
92     to << "cu_tpc_gas_FM-2" << endl;
93     to << "cu_tpc_gas_FM-3" << endl;
94     to << "cu_tpc_gas_FM-4" << endl;
95     to << "cu_tpc_gas_FM-5" << endl;
96     to << "cu_tpc_gas_FI-7" << endl;
97     to << "cu_tpc_gas_LEAK" << endl;
98     to << "cu_tpc_gas_CH4-FM" << endl;
99     //MPD added following three for Run_4
100    to << "cu_tpc_gas_PS-1" << endl;
```

```

101     to << "cu_tpc_gas_O2-M6" << endl;
102     to << "cu_tpc_gas_H2O-M7" << endl;
103
104
105     to.close();
106
107 #undef __METHOD__
108 } //

```

4.6.2.4 void tpcGasSender::initTable () [virtual]

Definition at line 29 of file tpcGasSender.cc.

```

29     {
30 #define __METHOD__ "initTable()"
31
32     StDbTable* table=0;
33     if(!(table=node->addDbTable("tpcGas")))
34         sendMess("Could not find table=tpcGas",dbmFatal,__LINE__,__CLASS__,__METHOD__);
35
36     memset(tempVals,0,NUM_DB_ROWS*sizeof(tpcGas));
37     memset(previousVals,0,NUM_DB_ROWS*sizeof(tpcGas));
38
39     int nrows;
40     int* elist = table->getElementID(nrows);
41     if(nrows!=NUM_DB_ROWS){
42         //char mess[256];
43         ostringstream ms;
44         ms<<"Db rows("<<nrows<<") != compiled("<<NUM_DB_ROWS<<")";
45         sendMess((ms.str()).c_str(),dbmFatal,__LINE__,__CLASS__,__METHOD__);
46     }
47     memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
48
49     unsigned int timestamp=time(NULL);
50     mgr->setRequestTime(timestamp);
51     if(mgr->fetchDbTable(table)){
52         tpcGas* thv = (tpcGas*)table->GetTable();
53         memcpy(previousVals,thv,nrows*sizeof(tpcGas));
54     }
55
56 #undef __METHOD__
57 };

```

4.6.2.5 void tpcGasSender::initTags () [virtual]

Definition at line 65 of file tpcGasSender.cc.

```

65     {
66     /* more than an example -> swap "trg" to your subsys & add to email list*/
67     setEmailTo("porter@bnl.gov");
68     setDomainName("tpc");
69
70 }

```

4.6.2.6 bool tpcGasSender::loadUserControls (const char * name, const char * value) [virtual]

Definition at line 20 of file tpcGasSender_i.cc.

```

20                                     {
21 #define __METHOD__ "loadUserControls(name,value)"
22
23 /* more than an example ... swap driftLimit to yours
24 * and duplicate this structure for each selection criteria
25 if(strstr(name,"driftLimit")){
26     driftLimit=atof(value);
27     sendMess("driftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
28     return true;
29 }
30 */
31
32 return false;
33 #undef __METHOD__
34 }

```

4.6.2.7 bool tpcGasSender::nextLine (ifstream & from) [inline]

Definition at line 74 of file tpcGasSender.hh.

```

74                                     {
75 if(!from.getline(mline,255))return false;
76 return true;
77 }

```

4.6.2.8 bool tpcGasSender::queryData () [virtual]

Definition at line 90 of file tpcGasSender.cc.

```

90                                     {
91 #define __METHOD__ "queryData()"
92
93 /*
94 * MORE THAN AN EXAMPLE...
95 * IF Standard SC-Query via "caGet" then,
96 * no need to change this method AT ALL
97 *
98 */
99
100 writeTime = (unsigned int)time(NULL); //for database write time
101
102 //char systemCmd[1024];
103 ostringstream scmd;
104 scmd<<"caGet "<<queryFile<<" "<<dataFile;
105
106 if(system((scmd.str()).c_str()))
107     return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
108
109 return true;
110 #undef __METHOD__
111 };

```

4.6.2.9 char * tpcGasSender::readAny ()

Definition at line 196 of file tpcGasSender.cc.

```

196         {
197
198     strcpy(tmpLine,mLine);
199     ptr1=tmpLine;
200     ptr2=strtok(ptr1," ");
201     if(!ptr2) return ptr2;
202     ptr2=strtok(NULL," ");
203     return ptr2;
204 }

```

4.6.2.10 bool tpcGasSender::readData(istream &from) [virtual]

Definition at line 113 of file tpcGasSender_i.cc.

```

113                                     {
114 #define __METHOD__ "readData(istream)"
115
116 mreadStatus=true;
117 memset(tempVals,0,NUM_DB_ROWS*sizeof(tpcGas));
118
119
120 int i=0;
121 char* c=__CLASS__;
122 char* m=__METHOD__;
123
124 /* example format
125 * for(int i=0;i<NUM_DB_ROWS;i++){
126 * if(!nextLine(from) || !readVal(tempVals[i].blah)) readError(l,c,m);
127 *
128 *     ....
129 *
130 * }
131 */
132 if(!nextLine(from) || !readVal(tempVals[0].CH4_P1)) readError(__LINE__,c,m);
133 if(!nextLine(from) || !readVal(tempVals[0].CH4_P2)) readError(__LINE__,c,m);
134 if(!nextLine(from) || !readVal(tempVals[0].O2_M1)) readError(__LINE__,c,m);
135 if(!nextLine(from) || !readVal(tempVals[0].O2_M5)) readError(__LINE__,c,m);
136 if(!nextLine(from) || !readVal(tempVals[0].CH4_M3)) readError(__LINE__,c,m);
137 if(!nextLine(from) || !readVal(tempVals[0].CH4_M4)) readError(__LINE__,c,m);
138 if(!nextLine(from) || !readVal(tempVals[0].PT_1)) readError(__LINE__,c,m);
139 if(!nextLine(from) || !readVal(tempVals[0].PT_2)) readError(__LINE__,c,m);
140 if(!nextLine(from) || !readVal(tempVals[0].PT_3)) readError(__LINE__,c,m);
141 // if(!nextLine(from) || !readVal(tempVals[0].PT_4)) readError(__LINE__,c,m);
142 if(!nextLine(from) || !readVal(tempVals[0].PT_5)) readError(__LINE__,c,m);
143 if(!nextLine(from) || !readVal(tempVals[0].PT_7)) readError(__LINE__,c,m);
144 if(!nextLine(from) || !readVal(tempVals[0].PT_8)) readError(__LINE__,c,m);
145 if(!nextLine(from) || !readVal(tempVals[0].PT_B)) readError(__LINE__,c,m);
146 // if(!nextLine(from) || !readVal(tempVals[0].PT_B2)) readError(__LINE__,c,m);
147 if(!nextLine(from) || !readVal(tempVals[0].PT_10)) readError(__LINE__,c,m);
148 if(!nextLine(from) || !readVal(tempVals[0].PT_9)) readError(__LINE__,c,m);
149 if(!nextLine(from) || !readVal(tempVals[0].PT_11)) readError(__LINE__,c,m);
150 if(!nextLine(from) || !readVal(tempVals[0].PI_8)) readError(__LINE__,c,m);
151 if(!nextLine(from) || !readVal(tempVals[0].PI_9)) readError(__LINE__,c,m);
152 if(!nextLine(from) || !readVal(tempVals[0].PI_10)) readError(__LINE__,c,m);
153 if(!nextLine(from) || !readVal(tempVals[0].PI_13)) readError(__LINE__,c,m);
154 if(!nextLine(from) || !readVal(tempVals[0].PI_14)) readError(__LINE__,c,m);
155 if(!nextLine(from) || !readVal(tempVals[0].PI_15)) readError(__LINE__,c,m);
156 if(!nextLine(from) || !readVal(tempVals[0].T1)) readError(__LINE__,c,m);
157 if(!nextLine(from) || !readVal(tempVals[0].T2)) readError(__LINE__,c,m);
158 if(!nextLine(from) || !readVal(tempVals[0].T3)) readError(__LINE__,c,m);
159 if(!nextLine(from) || !readVal(tempVals[0].T4)) readError(__LINE__,c,m);
160 if(!nextLine(from) || !readVal(tempVals[0].T5)) readError(__LINE__,c,m);
161 if(!nextLine(from) || !readVal(tempVals[0].T6)) readError(__LINE__,c,m);
162 if(!nextLine(from) || !readVal(tempVals[0].T7)) readError(__LINE__,c,m);

```

```

163 if(!nextLine(from) || !readVal(tempVals[0].T8)) readError(__LINE__,c,m);
164 if(!nextLine(from) || !readVal(tempVals[0].T9)) readError(__LINE__,c,m);
165 if(!nextLine(from) || !readVal(tempVals[0].T10)) readError(__LINE__,c,m);
166 if(!nextLine(from) || !readVal(tempVals[0].T11)) readError(__LINE__,c,m);
167 if(!nextLine(from) || !readVal(tempVals[0].T12)) readError(__LINE__,c,m);
168 if(!nextLine(from) || !readVal(tempVals[0].T13)) readError(__LINE__,c,m);
169 if(!nextLine(from) || !readVal(tempVals[0].T14)) readError(__LINE__,c,m);
170 if(!nextLine(from) || !readVal(tempVals[0].T15)) readError(__LINE__,c,m);
171 if(!nextLine(from) || !readVal(tempVals[0].T16)) readError(__LINE__,c,m);
172 if(!nextLine(from) || !readVal(tempVals[0].H2O_M2)) readError(__LINE__,c,m);
173 // if(!nextLine(from) || !readVal(tempVals[0].C2H6_M3)) readError(__LINE__,c,m);
174 if(!nextLine(from) || !readVal(tempVals[0].FM_1)) readError(__LINE__,c,m);
175 if(!nextLine(from) || !readVal(tempVals[0].FM_2)) readError(__LINE__,c,m);
176 if(!nextLine(from) || !readVal(tempVals[0].FM_3)) readError(__LINE__,c,m);
177 if(!nextLine(from) || !readVal(tempVals[0].FM_4)) readError(__LINE__,c,m);
178 if(!nextLine(from) || !readVal(tempVals[0].FM_5)) readError(__LINE__,c,m);
179 if(!nextLine(from) || !readVal(tempVals[0].FI_7)) readError(__LINE__,c,m);
180 if(!nextLine(from) || !readVal(tempVals[0].LEAK)) readError(__LINE__,c,m);
181 if(!nextLine(from) || !readVal(tempVals[0].CH4_FM)) readError(__LINE__,c,m);
182 if(!nextLine(from) || !readVal(tempVals[0].PS_1)) readError(__LINE__,c,m);
183 if(!nextLine(from) || !readVal(tempVals[0].O2_M6)) readError(__LINE__,c,m);
184 if(!nextLine(from) || !readVal(tempVals[0].H2O_M7)) readError(__LINE__,c,m);
185
186 from.close();
187 return true;
188 #undef __METHOD__
189 }

```

4.6.2.11 bool tpcGasSender::readData (const char * *fileName*) [virtual]

Definition at line 115 of file tpcGasSender.cc.

```

115                                     {
116 #define __METHOD__ "readData(fileName)"
117
118 ifstream from(fileName);
119 if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
120
121 return readData(from); // user implemented file read
122 #undef __METHOD__
123 }

```

4.6.2.12 void tpcGasSender::readError (int *l*, char * *c*, char * *m*) [inline]

Definition at line 79 of file tpcGasSender.hh.

```

79                                     {
80 mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
81 }

```

4.6.2.13 bool tpcGasSender::readVal (long long & *value*)

Definition at line 268 of file tpcGasSender.cc.

```

268                                     {
269
270 if(!readAny())return false;

```

```
271 char* store[256];
272 value=strtoll(ptr2,store,10);
273 if(strlen(*store)>0) return false; // value is not a number
274
275 return true;
276 };
```

4.6.2.14 bool tpcGasSender::readVal (long & value)

Definition at line 257 of file tpcGasSender.cc.

```
257                                     {
258
259     if(!readAny())return false;
260
261     char* store[256];
262     value=strtol(ptr2,store,10);
263     if(strlen(*store)>0) return false; // value is not a number
264
265     return true;
266 };
```

4.6.2.15 bool tpcGasSender::readVal (int & value)

Definition at line 246 of file tpcGasSender.cc.

```
246                                     {
247
248     if(!readAny()) return false;
249
250     char* store[256];
251     value=(int)strtol(ptr2,store,10);
252     if(strlen(*store)>0) return false; // value is not a number
253
254     return true;
255 };
```

4.6.2.16 bool tpcGasSender::readVal (short & value)

Definition at line 235 of file tpcGasSender.cc.

```
235                                     {
236
237     if(!readAny()) return false;
238
239     char* store[256];
240     value=(short)strtol(ptr2,store,10);
241     if(strlen(*store)>0) return false; // value is not a number
242
243     return true;
244 };
```

4.6.2.17 bool tpcGasSender::readVal (double & value)

Definition at line 224 of file tpcGasSender.cc.

```
224                                     {
225
226     if(!readAny())return false;
227
228     char* store[256];
229     value=strtod(ptr2,store);
230     if(strlen(*store)>0) return false; // value is not a number
231
232     return true;
233 };
```

4.6.2.18 bool tpcGasSender::readVal (float & value)

Definition at line 213 of file tpcGasSender.cc.

```
213                                     {
214
215     if(!readAny()) return false;
216
217     char* store[256];
218     value=(float)strtod(ptr2,store);
219     if(strlen(*store)>0) return false; // value is not a number
220
221     return true;
222 };
```

4.6.2.19 bool tpcGasSender::readVal (char *& value)

Definition at line 206 of file tpcGasSender.cc.

```
206                                     {
207
208     if(!readAny()) return false;
209     strcpy(value,ptr2);
210     return true;
211 }
```

4.6.2.20 bool tpcGasSender::updateDb (const char *fileName) [virtual]

Definition at line 126 of file tpcGasSender.cc.

```
126                                     {
127 #define __METHOD__ "updateDb(filename)"
128
129     if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
130
131     int* elements;
132     tpcGas* vals;
133     int numRows = 0;
134
135     if(writeRequired()){
136
```

```

137     numRows=NUM_DB_ROWS;
138     elements=elementList;
139     vals = tempVals;
140
141 } else {
142
143     for(int i=0; i<NUM_DB_ROWS; i++){
144         if(hasChanged(i)){
145             updateElements[numRows]=elementList[i];
146             updateVals[numRows] = tempVals[i];
147             previousVals[i]=tempVals[i];
148             numRows++;
149         }
150     }
151
152     elements = updateElements;
153     vals     = updateVals;
154 }
155
156 if(numRows==0) return sendMess(" No update required for",mbaseName,dbMDebug,__LINE__,__CLASS__,__MET
157
158 //char mess[256];
159 ostreamstream sn;
160 sn<<"Will Update " <<numRows<<" of " <<NUM_DB_ROWS<<" rows ";
161 sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
162
163 StDbTable* dbTable=node->findTable("tpcGas");
164 dbTable->SetTable((char*)vals, numRows, elements);
165 mgr->setStoreTime(writeTime);
166
167 if(!mgr->storeDbTable(dbTable)) {
168     addBackLog(writeTime);
169     return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
170 }
171
172 if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
173
174 return true;
175 #undef __METHOD__
176 }

```

4.6.3 Member Data Documentation

4.6.3.1 int [tpcGasSender::elementList](#)[NUM_DB_ROWS] [protected]

Definition at line 24 of file tpcGasSender.hh.

4.6.3.2 char [tpcGasSender::mline](#)[256] [protected]

Definition at line 29 of file tpcGasSender.hh.

4.6.3.3 bool [tpcGasSender::mreadStatus](#) [protected]

Definition at line 28 of file tpcGasSender.hh.

4.6.3.4 tpcGas [tpcGasSender::previousVals](#)[NUM_DB_ROWS] [protected]

Definition at line 22 of file tpcGasSender.hh.

4.6.3.5 `char* tpcGasSender::ptr1` [protected]

Definition at line 31 of file tpcGasSender.hh.

4.6.3.6 `char * tpcGasSender::ptr2` [protected]

Definition at line 31 of file tpcGasSender.hh.

4.6.3.7 `tpcGas tpcGasSender::tempVals[NUM_DB_ROWS]` [protected]

Definition at line 23 of file tpcGasSender.hh.

4.6.3.8 `char tpcGasSender::tmpline[256]` [protected]

Definition at line 30 of file tpcGasSender.hh.

4.6.3.9 `int tpcGasSender::updateElements[NUM_DB_ROWS]` [protected]

Definition at line 26 of file tpcGasSender.hh.

4.6.3.10 `tpcGas tpcGasSender::updateVals[NUM_DB_ROWS]` [protected]

Definition at line 25 of file tpcGasSender.hh.

The documentation for this class was generated from the following files:

- [tpcGasSender.hh](#)
- [tpcGasSender.cc](#)
- [tpcGasSender_i.cc](#)

4.7 tpcISANodeSender Class Reference

```
#include <tpcISANodeSender.hh>
```

Public Member Functions

- [tpcISANodeSender](#) (const char *localDir)
- virtual [~tpcISANodeSender](#) ()
- virtual void [initTable](#) ()
- virtual void [initTags](#) ()
- virtual void [initDataBase](#) ()
- virtual bool [loadUserControls](#) (const char *name, const char *value)
- virtual void [initQuery](#) ()
- virtual bool [queryData](#) ()
- virtual bool [readData](#) (const char *fileName)
- virtual bool [updateDb](#) (const char *fileName)
- virtual bool [readData](#) (ifstream &from)
- virtual bool [hasChanged](#) (int rowNumber)
- char * [readAny](#) ()
- bool [readVal](#) (char *&value)
- bool [readVal](#) (float &value)
- bool [readVal](#) (double &value)
- bool [readVal](#) (short &value)
- bool [readVal](#) (int &value)
- bool [readVal](#) (long &value)
- bool [readVal](#) (long long &value)
- bool [nextLine](#) (ifstream &from)
- void [readError](#) (int l, char *c, char *m)

Protected Attributes

- tpcISANode [previousVals](#) [NUM_DB_ROWS]
- tpcISANode [tempVals](#) [NUM_DB_ROWS]
- int [elementList](#) [NUM_DB_ROWS]
- tpcISANode [updateVals](#) [NUM_DB_ROWS]
- int [updateElements](#) [NUM_DB_ROWS]
- bool [mreadStatus](#)
- char [mline](#) [256]
- char [tmpline](#) [256]
- char * [ptr1](#)
- char * [ptr2](#)
- float [vdriftLimit](#)
ditto
- float [cdriftLimit](#)

4.7.1 Constructor & Destructor Documentation

4.7.1.1 tpcISANodeSender::tpcISANodeSender (const char * localDir)

Definition at line 19 of file tpcISANodeSender.cc.

```

19                                     {
20
21     initTags();
22     if(localDir) cd(localDir);// note this ignores the sub dir tag
23     init("tpcISANode"); // setup the file I/O
24     initDataBase();          // database connections
25     initTable();             // table definitions
26
27 }
```

4.7.1.2 virtual tpcISANodeSender::~tpcISANodeSender () [inline, virtual]

Definition at line 42 of file tpcISANodeSender.hh.

```
42 {};
```

4.7.2 Member Function Documentation

4.7.2.1 bool tpcISANodeSender::hasChanged (int rowNumber) [virtual]

Definition at line 146 of file tpcISANodeSender_i.cc.

```

146                                     {
147
148     tpcISANode* pre=&previousVals[rowNumber];
149     tpcISANode* cur=&tempVals[rowNumber];
150
151     if(fabs(pre->mVolt1-cur->mVolt1)>=vdriftLimit) return true;
152     if(fabs(pre->mVolt2-cur->mVolt2)>=vdriftLimit) return true;
153     if(fabs(pre->mVolt3-cur->mVolt3)>=vdriftLimit) return true;
154     if(fabs(pre->mVolt4-cur->mVolt4)>=vdriftLimit) return true;
155     if(fabs(pre->mCurrent1-cur->mCurrent1)>=cdriftLimit) return true;
156     if(fabs(pre->mCurrent2-cur->mCurrent2)>=cdriftLimit) return true;
157     if(fabs(pre->mCurrent3-cur->mCurrent3)>=cdriftLimit) return true;
158     if(fabs(pre->mCurrent4-cur->mCurrent4)>=cdriftLimit) return true;
159
160     /* example ... note -> change to any element requires db-update
161     * and thus returns true immediately
162     *
163     *if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
164     *if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
165     *
166     * ....
167     */
168
169     return false;
170 }
```

4.7.2.2 void tpcISANodeSender::initDataBase () [virtual]

Definition at line 75 of file tpcISANodeSender.cc.

```

75                                     {
76 #define __METHOD__ "initDataBase()"
77
78 /* More than an example... swap user & dbTrg as per subsystem*/
79 mgr->setUser("stardb","");
80 StDbType dbT = dbConditions;
81 StDbDomain dbD = dbTpc;
82
83 if( !( node = mgr->initConfig(dbT,dbD) ) )
84     sendMess("Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal,__LINE__,__CLASS__,__METHOD__);
85
86 #undef __METHOD__
87 }

```

4.7.2.3 void tpcISANodeSender::initQuery () [virtual]

Definition at line 48 of file tpcISANodeSender_i.cc.

```

48                                     {
49 #define __METHOD__ "initQuery()"
50
51     ofstream to(queryFile);
52
53     if(!to.is_open()){
54         sendMess("Open Failed ",queryFile,dbMFatal,__LINE__,__CLASS__,__METHOD__);
55         return;
56     }
57
58
59     const char *bar[8] = {"E","F","G","H","I","J","K","L"};
60     int i=1; // innersector
61
62     for(int j = 0; j < 12; j++){ // card number
63         for(int k = 0; k < 8; k++){ //channel number
64             to<<"tpchv:MC_"<<i<<"_"<<j<<"_"<<k<<endl; //measured current
65             to<<"tpchv:SUB_RD_MV_"<<i<<"_"<<j<<"_"<<bar[k]<<endl; // measured volt
66             to<<"tpchv:SUB_RD_ST_"<<i<<"_"<<j<<"_"<<bar[k]<<endl; //trip status
67             to<<"tpchv:SUB_RD_CE_"<<i<<"_"<<j<<"_"<<bar[k]<<endl; //enabled
68             to<<"tpchv:SUB_HV_"<<i<<"_"<<j<<"_"<<bar[k]<<endl; //on
69         }
70     }
71
72 /* example
73 *     for(int i=0;i<16;i++){
74 *         to<<"TRGhv:SUB_RD_V_1:"<<i<<".E"<<endl;
75 *         to<<"TRGhv:SUB_RD_V_1:"<<i<<".F"<<endl;
76 *         ....
77 *     }
78 */
79
80     to.close();
81
82
83 #undef __METHOD__
84 }

```

4.7.2.4 void tpcISANodeSender::initTable () [virtual]

Definition at line 30 of file tpcISANodeSender.cc.

```

30                                     {

```

```

31 #define __METHOD__ "initTable()"
32
33     StDbTable* table=0;
34     if(!(table=node->addDbTable("tpcISANode")))
35         sendMess("Could not find table=tpcISANode",dbmFatal,__LINE__,__CLASS__,__METHOD__);
36
37     memset(tempVals,0,NUM_DB_ROWS*sizeof(tpcISANode));
38     memset(previousVals,0,NUM_DB_ROWS*sizeof(tpcISANode));
39
40     int nrows;
41     int* elist = table->getElementID(nrows);
42     if(nrows!=NUM_DB_ROWS){
43         //char mess[256];
44         ostringstream ms;
45         ms<<"Db rows("<<nrows<<" != compiled("<<NUM_DB_ROWS<<")";
46         sendMess((ms.str()).c_str(),dbmFatal,__LINE__,__CLASS__,__METHOD__);
47     }
48     memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
49
50     unsigned int timestamp=time(NULL);
51     mgr->setRequestTime(timestamp);
52     if(mgr->fetchDbTable(table)){
53         tpcISANode* thv = (tpcISANode*)table->GetTable();
54         memcpy(previousVals,thv,nrows*sizeof(tpcISANode));
55     }
56
57 #undef __METHOD__
58 };

```

4.7.2.5 void tpcISANodeSender::initTags () [virtual]

Definition at line 66 of file tpcISANodeSender.cc.

```

66     {
67     /* more than an example -> swap "trg" to your subsys & add to email list*/
68     setEmailTo("porter@bnl.gov");
69     setDomainName("tpc");
70
71 }

```

4.7.2.6 bool tpcISANodeSender::loadUserControls (const char * name, const char * value) [virtual]

Definition at line 20 of file tpcISANodeSender_i.cc.

```

20     {
21 #define __METHOD__ "loadUserControls(name,value)"
22
23 /* more than an example ... swap driftLimit to yours
24 * and duplicate this structure for each selection criteria
25 if(strstr(name,"driftLimit")){
26     driftLimit=atof(value);
27     sendMess("driftLimit set=",value,dbmDebug,__LINE__,__CLASS__,__METHOD__);
28     return true;
29 }
30 */
31 if(strstr(name,"vdriftLimit")){
32     vdriftLimit=atof(value);
33     sendMess("vdriftLimit set=",value,dbmDebug,__LINE__,__CLASS__,__METHOD__);
34     return true;

```

```

35 }
36 if(strstr(name,"cdriftLimit")){
37     cdriftLimit=atof(value);
38     sendMess("cdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
39     return true;
40 }
41
42 return false;
43 #undef __METHOD__
44 }

```

4.7.2.7 bool tpcISAnodeSender::nextLine (ifstream &from) [inline]

Definition at line 75 of file tpcISAnodeSender.hh.

```

75                                     {
76     if(!from.getline(mline,255))return false;
77     return true;
78 }

```

4.7.2.8 bool tpcISAnodeSender::queryData () [virtual]

Definition at line 91 of file tpcISAnodeSender.cc.

```

91                                     {
92 #define __METHOD__ "queryData()"
93
94 /*
95  * MORE THAN AN EXAMPLE...
96  * IF Standard SC-Query via "caGet" then,
97  * no need to change this method AT ALL
98  *
99  */
100
101     writeTime = (unsigned int)time(NULL); //for database write time
102
103     //char systemCmd[1024];
104     ostringstream scmd;
105     scmd<<"caGet "<<queryFile<<" "<<dataFile;
106
107     if(system((scmd.str()).c_str()))
108         return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
109
110     return true;
111 #undef __METHOD__
112 };

```

4.7.2.9 char * tpcISAnodeSender::readAny ()

Definition at line 197 of file tpcISAnodeSender.cc.

```

197                                     {
198
199     strcpy(tmpline,mline);
200     ptr1=tmpline;
201     ptr2=strtok(ptr1," ");
202     if(!ptr2) return ptr2;

```

```

203 ptr2=strtok(NULL, " ");
204 return ptr2;
205 }

```

4.7.2.10 bool tpcISANodeSender::readData (ifstream & from) [virtual]

Definition at line 89 of file tpcISANodeSender_i.cc.

```

89                                     {
90 #define __METHOD__ "readData(ifstream)"
91
92 mreadStatus=true;
93 memset(tempVals,0,NUM_DB_ROWS*sizeof(tpcISANode));
94
95
96 //int i=0;
97 char* c=__CLASS__;
98 char* m=__METHOD__;
99
100 for(int i=0;i<24;i++){
101
102     if(!nextLine(from) || !readVal(tempVals[i].mVolt4)) readError(__LINE__,c,m);
103     if(!nextLine(from) || !readVal(tempVals[i].mCurrent4)) readError(__LINE__,c,m);
104     if(!nextLine(from) || !readVal(tempVals[i].trip4)) readError(__LINE__,c,m);
105     if(!nextLine(from) || !readVal(tempVals[i].enable4)) readError(__LINE__,c,m);
106     if(!nextLine(from) || !readVal(tempVals[i].channelOn4)) readError(__LINE__,c,m);
107
108     if(!nextLine(from) || !readVal(tempVals[i].mVolt3)) readError(__LINE__,c,m);
109     if(!nextLine(from) || !readVal(tempVals[i].mCurrent3)) readError(__LINE__,c,m);
110     if(!nextLine(from) || !readVal(tempVals[i].trip3)) readError(__LINE__,c,m);
111     if(!nextLine(from) || !readVal(tempVals[i].enable3)) readError(__LINE__,c,m);
112     if(!nextLine(from) || !readVal(tempVals[i].channelOn3)) readError(__LINE__,c,m);
113
114     if(!nextLine(from) || !readVal(tempVals[i].mVolt2)) readError(__LINE__,c,m);
115     if(!nextLine(from) || !readVal(tempVals[i].mCurrent2)) readError(__LINE__,c,m);
116     if(!nextLine(from) || !readVal(tempVals[i].trip2)) readError(__LINE__,c,m);
117     if(!nextLine(from) || !readVal(tempVals[i].enable2)) readError(__LINE__,c,m);
118     if(!nextLine(from) || !readVal(tempVals[i].channelOn2)) readError(__LINE__,c,m);
119
120     if(!nextLine(from) || !readVal(tempVals[i].mVolt1)) readError(__LINE__,c,m);
121     if(!nextLine(from) || !readVal(tempVals[i].mCurrent1)) readError(__LINE__,c,m);
122     if(!nextLine(from) || !readVal(tempVals[i].trip1)) readError(__LINE__,c,m);
123     if(!nextLine(from) || !readVal(tempVals[i].enable1)) readError(__LINE__,c,m);
124     if(!nextLine(from) || !readVal(tempVals[i].channelOn1)) readError(__LINE__,c,m);
125 }
126
127
128
129
130 /* example format
131 * for(int i=0;i<NUM_DB_ROWS;i++){
132 * if(!nextLine(from) || !readVal(tempVals[i].blah)) readError(l,c,m);
133 *
134 *     ....
135 *
136 * }
137 */
138
139 from.close();
140 return true;
141 #undef __METHOD__
142 }

```

4.7.2.11 bool tpcISANodeSender::readData (const char **fileName*) [virtual]

Definition at line 116 of file tpcISANodeSender.cc.

```
116                                     {
117 #define __METHOD__ "readData(fileName)"
118
119     ifstream from(fileName);
120     if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
121
122     return readData(from); // user implemented file read
123 #undef __METHOD__
124 }
```

4.7.2.12 void tpcISANodeSender::readError (int *l*, char **c*, char **m*) [inline]

Definition at line 80 of file tpcISANodeSender.hh.

```
80                                     {
81     mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
82 }
```

4.7.2.13 bool tpcISANodeSender::readVal (long long & *value*)

Definition at line 269 of file tpcISANodeSender.cc.

```
269                                     {
270
271     if(!readAny())return false;
272     char* store[256];
273     value=strtoll(ptr2,store,10);
274     if(strlen(*store)>0) return false; // value is not a number
275
276     return true;
277 };
```

4.7.2.14 bool tpcISANodeSender::readVal (long & *value*)

Definition at line 258 of file tpcISANodeSender.cc.

```
258                                     {
259
260     if(!readAny())return false;
261
262     char* store[256];
263     value=strtol(ptr2,store,10);
264     if(strlen(*store)>0) return false; // value is not a number
265
266     return true;
267 };
```

4.7.2.15 bool tpcISANodeSender::readVal (int & value)

Definition at line 247 of file tpcISANodeSender.cc.

```
247                                     {
248
249     if(!readAny()) return false;
250
251     char* store[256];
252     value=(int)strtol(ptr2,store,10);
253     if(strlen(*store)>0) return false; // value is not a number
254
255     return true;
256 };
```

4.7.2.16 bool tpcISANodeSender::readVal (short & value)

Definition at line 236 of file tpcISANodeSender.cc.

```
236                                     {
237
238     if(!readAny()) return false;
239
240     char* store[256];
241     value=(short)strtol(ptr2,store,10);
242     if(strlen(*store)>0) return false; // value is not a number
243
244     return true;
245 };
```

4.7.2.17 bool tpcISANodeSender::readVal (double & value)

Definition at line 225 of file tpcISANodeSender.cc.

```
225                                     {
226
227     if(!readAny())return false;
228
229     char* store[256];
230     value=strtod(ptr2,store);
231     if(strlen(*store)>0) return false; // value is not a number
232
233     return true;
234 };
```

4.7.2.18 bool tpcISANodeSender::readVal (float & value)

Definition at line 214 of file tpcISANodeSender.cc.

```
214                                     {
215
216     if(!readAny()) return false;
217
218     char* store[256];
219     value=(float)strtod(ptr2,store);
220     if(strlen(*store)>0) return false; // value is not a number
221
222     return true;
223 };
```

4.7.2.19 bool tpcISANodeSender::readVal (char *& value)

Definition at line 207 of file tpcISANodeSender.cc.

```

207                                     {
208
209   if(!readAny()) return false;
210   strcpy(value,ptr2);
211   return true;
212 }
```

4.7.2.20 bool tpcISANodeSender::updateDb (const char *fileName) [virtual]

Definition at line 127 of file tpcISANodeSender.cc.

```

127                                     {
128 #define __METHOD__ "updateDb(filename)"
129
130   if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
131
132   int* elements;
133   tpcISANode* vals;
134   int numRows = 0;
135
136   if(writeRequired()){
137
138     numRows=NUM_DB_ROWS;
139     elements=elementList;
140     vals = tempVals;
141
142   } else {
143
144     for(int i=0; i<NUM_DB_ROWS; i++){
145       if(hasChanged(i)){
146         updateElements[numRows]=elementList[i];
147         updateVals[numRows] = tempVals[i];
148         previousVals[i]=tempVals[i];
149         numRows++;
150       }
151     }
152
153     elements = updateElements;
154     vals     = updateVals;
155   }
156
157   if(numRows==0) return sendMess(" No update required for",mbaseName,dbMDebug,__LINE__,__CLASS__,__METHOD__);
158
159   //char mess[256];
160   ostreamstream sn;
161   sn<<"Will Update "<<numRows<<" of "<<NUM_DB_ROWS<<" rows ";
162   sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
163
164   StDbTable* dbTable=node->findTable("tpcISANode");
165   dbTable->SetTable((char*)vals, numRows, elements);
166   mgr->setStoreTime(writeTime);
167
168   if(!mgr->storeDbTable(dbTable)) {
169     addBackLog(writeTime);
170     return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
171   }
172
173   if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
174
```

```
175     return true;
176     #undef __METHOD__
177 }
```

4.7.3 Member Data Documentation

4.7.3.1 float [tpcISANodeSender::cdriftLimit](#) [protected]

Definition at line 36 of file tpcISANodeSender.hh.

4.7.3.2 int [tpcISANodeSender::elementList](#)[NUM_DB_ROWS] [protected]

Definition at line 24 of file tpcISANodeSender.hh.

4.7.3.3 char [tpcISANodeSender::mline](#)[256] [protected]

Definition at line 29 of file tpcISANodeSender.hh.

4.7.3.4 bool [tpcISANodeSender::mreadStatus](#) [protected]

Definition at line 28 of file tpcISANodeSender.hh.

4.7.3.5 tpcISANode [tpcISANodeSender::previousVals](#)[NUM_DB_ROWS] [protected]

Definition at line 22 of file tpcISANodeSender.hh.

4.7.3.6 char* [tpcISANodeSender::ptr1](#) [protected]

Definition at line 31 of file tpcISANodeSender.hh.

4.7.3.7 char * [tpcISANodeSender::ptr2](#) [protected]

Definition at line 31 of file tpcISANodeSender.hh.

4.7.3.8 tpcISANode [tpcISANodeSender::tempVals](#)[NUM_DB_ROWS] [protected]

Definition at line 23 of file tpcISANodeSender.hh.

4.7.3.9 char [tpcISANodeSender::tmpline](#)[256] [protected]

Definition at line 30 of file tpcISANodeSender.hh.

4.7.3.10 int [tpcISANodeSender::updateElements](#)[NUM_DB_ROWS] [protected]

Definition at line 26 of file tpcISANodeSender.hh.

4.7.3.11 `tpcISAnode` `tpcISAnodeSender::updateVals[NUM_DB_ROWS]` [protected]

Definition at line 25 of file `tpcISAnodeSender.hh`.

4.7.3.12 `float` `tpcISAnodeSender::vdriftLimit` [protected]

dito

Definition at line 35 of file `tpcISAnodeSender.hh`.

The documentation for this class was generated from the following files:

- [tpcISAnodeSender.hh](#)
- [tpcISAnodeSender.cc](#)
- [tpcISAnodeSender_i.cc](#)

4.8 tpcISGGridSender Class Reference

```
#include <tpcISGGridSender.h>
```

Public Member Functions

- [tpcISGGridSender](#) (const char *localDir)
- virtual [~tpcISGGridSender](#) ()
- virtual void [initTable](#) ()
- virtual void [initTags](#) ()
- virtual void [initDataBase](#) ()
- virtual bool [loadUserControls](#) (const char *name, const char *value)
- virtual void [initQuery](#) ()
- virtual bool [queryData](#) ()
- virtual bool [readData](#) (const char *fileName)
- virtual bool [updateDb](#) (const char *fileName)
- virtual bool [readData](#) (ifstream &from)
- virtual bool [hasChanged](#) (int rowNumber)
- char * [readAny](#) ()
- bool [readVal](#) (char *&value)
- bool [readVal](#) (float &value)
- bool [readVal](#) (double &value)
- bool [readVal](#) (short &value)
- bool [readVal](#) (int &value)
- bool [readVal](#) (long &value)
- bool [readVal](#) (long long &value)
- bool [nextLine](#) (ifstream &from)
- void [readError](#) (int l, char *c, char *m)

Protected Attributes

- tpcISGGrid [previousVals](#) [NUM_DB_ROWS]
- tpcISGGrid [tempVals](#) [NUM_DB_ROWS]
- int [elementList](#) [NUM_DB_ROWS]
- tpcISGGrid [updateVals](#) [NUM_DB_ROWS]
- int [updateElements](#) [NUM_DB_ROWS]
- bool [mreadStatus](#)
- char [mline](#) [256]
- char [tmpline](#) [256]
- char * [ptr1](#)
- char * [ptr2](#)
- float [vdriftLimit](#)

ditto

4.8.1 Constructor & Destructor Documentation

4.8.1.1 `tpcISGGridSender::tpcISGGridSender (const char * localDir)`

Definition at line 19 of file `tpcISGGridSender.cc`.

```

19                                     {
20
21     initTags();
22     if(localDir) cd(localDir);// note this ignores the sub dir tag
23     init("tpcISGGrid"); // setup the file I/O
24     initDataBase();      // database connections
25     initTable();         // table definitions
26
27 }
```

4.8.1.2 `virtual tpcISGGridSender::~~tpcISGGridSender ()` [inline, virtual]

Definition at line 41 of file `tpcISGGridSender.hh`.

```
41 {};
```

4.8.2 Member Function Documentation

4.8.2.1 `bool tpcISGGridSender::hasChanged (int rowNumber)` [virtual]

Definition at line 149 of file `tpcISGGridSender_i.cc`.

```

149                                     {
150
151     tpcISGGrid* pre=&previousVals[rowNumber];
152     tpcISGGrid* cur=&tempVals[rowNumber];
153
154     if(fabs(pre->refVolt-cur->refVolt)>=vdriftLimit) return true;
155     if(fabs(pre->hiVolt-cur->hiVolt)>=vdriftLimit) return true;
156     if(fabs(pre->loVolt-cur->loVolt)>=vdriftLimit) return true;
157
158     /* example ... note -> change to any element requires db-update
159      * and thus returns true immediately
160      *
161      *if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
162      *if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
163      *
164      * ....
165      */
166
167     return false;
168 }
```

4.8.2.2 `void tpcISGGridSender::initDataBase ()` [virtual]

Definition at line 75 of file `tpcISGGridSender.cc`.

```

75                                     {
76 #define __METHOD__ "initDataBase()"
```

```

77
78  /* More than an example... swap user & dbTrg as per subsystem*/
79  mgr->setUser("stardb","");
80  StDbType   dbT = dbConditions;
81  StDbDomain dbD = dbTpc;
82
83  if( !( node = mgr->initConfig(dbT,dbD)) )
84      sendMess("Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal, __LINE__, __CLASS__, __METHOD__);
85
86 #undef  __METHOD__
87 }

```

4.8.2.3 void tpcISGGridSender::initQuery () [virtual]

Definition at line 38 of file tpcISGGridSender_i.cc.

```

38  {
39 #define  __METHOD__  "initQuery()"
40
41      ofstream to(queryFile);
42
43      if(!to.is_open()){
44          sendMess("Open Failed ",queryFile,dbMFatal, __LINE__, __CLASS__, __METHOD__);
45          return;
46      }
47
48      int i;
49      for(i = 12; i <24; i++){
50          to<<"GG_A_In_A_"<<i+1<<endl;
51          to<<"GG_A_In_B_"<<i+1<<endl;
52          to<<"GG_A_In_C_"<<i+1<<endl;
53          to<<"GG_D_Out_D_"<<i+1<<endl;
54          to<<"GG_D_Out_B_"<<i+1<<endl;
55          to<<"GG_D_Out_C_"<<(i/4 + 1)*4<<endl;
56          to<<"GG_D_Out_A_"<<(i/4 + 1)*4<<endl;
57      }
58
59      for(i = 36; i <48; i++){
60          to<<"GG_A_In_A_"<<i+1<<endl;
61          to<<"GG_A_In_B_"<<i+1<<endl;
62          to<<"GG_A_In_C_"<<i+1<<endl;
63          to<<"GG_D_Out_D_"<<i+1<<endl;
64          to<<"GG_D_Out_B_"<<i+1<<endl;
65          to<<"GG_D_Out_C_"<<(i/4 + 1)*4<<endl;
66          to<<"GG_D_Out_A_"<<(i/4 + 1)*4<<endl;
67      }
68
69  /* example
70  *   for(int i=0;i<16;i++){
71  *       to<<"TRGhv:SUB_RD_V_1:"<<i<<".E"<<endl;
72  *       to<<"TRGhv:SUB_RD_V_1:"<<i<<".F"<<endl;
73  *       ....
74  *   }
75  */
76
77      to.close();
78
79
80 #undef  __METHOD__
81 }

```

4.8.2.4 void tpcISGGridSender::initTable () [virtual]

Definition at line 30 of file tpcISGGridSender.cc.

```

30                                     {
31 #define __METHOD__ "initTable()"
32
33     StDbTable* table=0;
34     if(!(table=node->addDbTable("tpcISGGrid")))
35         sendMess("Could not find table=tpcISGGrid",dbmFatal,__LINE__,__CLASS__,__METHOD__);
36
37     memset(tempVals,0,NUM_DB_ROWS*sizeof(tpcISGGrid));
38     memset(previousVals,0,NUM_DB_ROWS*sizeof(tpcISGGrid));
39
40     int nrows;
41     int* elist = table->getElementID(nrows);
42     if(nrows!=NUM_DB_ROWS){
43         // char mess[256];
44         ostringstream ms;
45         ms<<"Db rows("<<nrows<<") != compiled("<<NUM_DB_ROWS<<")";
46         sendMess((ms.str()).c_str(),dbmFatal,__LINE__,__CLASS__,__METHOD__);
47     }
48     memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
49
50     unsigned int timestamp=time(NULL);
51     mgr->setRequestTime(timestamp);
52     if(mgr->fetchDbTable(table)){
53         tpcISGGrid* thv = (tpcISGGrid*)table->GetTable();
54         memcpy(previousVals,thv,nrows*sizeof(tpcISGGrid));
55     }
56
57 #undef __METHOD__
58 };

```

4.8.2.5 void tpcISGGridSender::initTags () [virtual]

Definition at line 66 of file tpcISGGridSender.cc.

```

66                                     {
67 /* more than an example -> swap "trg" to your subsys & add to email list*/
68     setEmailTo("porter@bnl.gov");
69     setDomainName("tpc");
70
71 }

```

4.8.2.6 bool tpcISGGridSender::loadUserControls (const char * name, const char * value) [virtual]

Definition at line 20 of file tpcISGGridSender_i.cc.

```

20                                     {
21 #define __METHOD__ "loadUserControls(name,value)"
22
23 /* more than an example ... swap driftLimit to yours
24 * and duplicate this structure for each selection criteria */
25     if(strstr(name,"vdriftLimit")){
26         vdriftLimit=atof(value);
27         sendMess("vdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
28         return true;

```

```

29  }
30
31
32 return false;
33 #undef __METHOD__
34 }

```

4.8.2.7 bool tpcISGGridSender::nextLine (ifstream &from) [inline]

Definition at line 74 of file tpcISGGridSender.hh.

```

74                                     {
75   if(!from.getline(mline,255))return false;
76   return true;
77 }

```

4.8.2.8 bool tpcISGGridSender::queryData () [virtual]

Definition at line 91 of file tpcISGGridSender.cc.

```

91                                     {
92 #define __METHOD__ "queryData()"
93
94 /*
95  * MORE THAN AN EXAMPLE...
96  * IF Standard SC-Query via "caGet" then,
97  * no need to change this method AT ALL
98  *
99 */
100
101 writeTime = (unsigned int)time(NULL); //for database write time
102
103 //char systemCmd[1024];
104 ostreamstream scmd;
105 scmd<<"caGet "<<queryFile<<" "<<dataFile;
106
107 if(system((scmd.str()).c_str()));
108   return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
109
110 return true;
111 #undef __METHOD__
112 };

```

4.8.2.9 char * tpcISGGridSender::readAny ()

Definition at line 197 of file tpcISGGridSender.cc.

```

197                                     {
198
199   strcpy(tmpline,mline);
200   ptr1=tmpline;
201   ptr2=strtok(ptr1," ");
202   if(!ptr2) return ptr2;
203   ptr2=strtok(NULL," ");
204   return ptr2;
205 }

```

4.8.2.10 bool tpcISGGridSender::readData (ifstream & from) [virtual]

Definition at line 86 of file tpcISGGridSender_i.cc.

```

86                                     {
87 #define __METHOD__ "readData(ifstream)"
88
89 mreadStatus=true;
90 memset(tempVals,0,NUM_DB_ROWS*sizeof(tpcISGGrid));
91
92
93 //int i=0;
94 char* c=__CLASS__;
95 char* m=__METHOD__;
96
97 for(int i=0;i<24;i++){
98     if(!nextLine(from) || !readVal(tempVals[i].refVolt)) readError(__LINE__,c,m);
99     if(!nextLine(from) || !readVal(tempVals[i].hiVolt)) readError(__LINE__,c,m);
100    if(!nextLine(from) || !readVal(tempVals[i].loVolt)) readError(__LINE__,c,m);
101 //MPD Changes
102    char* tmpval0;
103    if(!nextLine(from)){
104        tmpval0=tempVals[i].hvEnable;
105        if(!readVal(tmpval0))readError(__LINE__,c,m);
106    }
107
108
109 //    if(!nextLine(from) || !readVal(tempVals[i].hvEnable)) readError(__LINE__,c,m);
110
111    char* tmpvall;
112
113    if(!nextLine(from)){
114        tmpvall=tempVals[i].psEnable;
115        if(!readVal(tmpvall))readError(__LINE__,c,m);
116    }
117    // if(!nextLine(from) || !readVal(tempVals[i].psEnable)) readError(__LINE__,c,m);
118    char* tmpval2;
119
120    if(!nextLine(from)){
121        tmpval2=tempVals[i].GateEnable;
122        if(!readVal(tmpval2))readError(__LINE__,c,m);
123    }
124    // if(!nextLine(from) || !readVal(tempVals[i].GateEnable)) readError(__LINE__,c,m);
125    char* tmpval3;
126
127    if(!nextLine(from)){
128        tmpval3=tempVals[i].trigEnable;
129        if(!readVal(tmpval3))readError(__LINE__,c,m);
130    }
131    //if(!nextLine(from) || !readVal(tempVals[i].trigEnable)) readError(__LINE__,c,m);
132 }
133 /* example format
134 * for(int i=0;i<NUM_DB_ROWS;i++){
135 * if(!nextLine(from) || !readVal(tempVals[i].blah)) readError(l,c,m);
136 *
137 *     ....
138 *
139 * }
140 */
141
142 from.close();
143 return true;
144 #undef __METHOD__
145 }

```

4.8.2.11 bool tpcISGGridSender::readData (const char * *fileName*) [virtual]

Definition at line 116 of file tpcISGGridSender.cc.

```
116                                     {
117 #define __METHOD__ "readData(fileName)"
118
119     ifstream from(fileName);
120     if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
121
122     return readData(from); // user implemented file read
123 #undef __METHOD__
124 }
```

4.8.2.12 void tpcISGGridSender::readError (int *l*, char * *c*, char * *m*) [inline]

Definition at line 79 of file tpcISGGridSender.hh.

```
79                                     {
80     mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
81 }
```

4.8.2.13 bool tpcISGGridSender::readVal (long long & *value*)

Definition at line 269 of file tpcISGGridSender.cc.

```
269                                     {
270
271     if(!readAny())return false;
272     char* store[256];
273     value=strtoll(ptr2,store,10);
274     if(strlen(*store)>0) return false; // value is not a number
275
276     return true;
277 };
```

4.8.2.14 bool tpcISGGridSender::readVal (long & *value*)

Definition at line 258 of file tpcISGGridSender.cc.

```
258                                     {
259
260     if(!readAny())return false;
261
262     char* store[256];
263     value=strtoll(ptr2,store,10);
264     if(strlen(*store)>0) return false; // value is not a number
265
266     return true;
267 };
```

4.8.2.15 bool tpcISGGridSender::readVal (int & value)

Definition at line 247 of file tpcISGGridSender.cc.

```
247                                     {
248
249     if(!readAny()) return false;
250
251     char* store[256];
252     value=(int)strtol(ptr2,store,10);
253     if(strlen(*store)>0) return false; // value is not a number
254
255     return true;
256 };
```

4.8.2.16 bool tpcISGGridSender::readVal (short & value)

Definition at line 236 of file tpcISGGridSender.cc.

```
236                                     {
237
238     if(!readAny()) return false;
239
240     char* store[256];
241     value=(short)strtol(ptr2,store,10);
242     if(strlen(*store)>0) return false; // value is not a number
243
244     return true;
245 };
```

4.8.2.17 bool tpcISGGridSender::readVal (double & value)

Definition at line 225 of file tpcISGGridSender.cc.

```
225                                     {
226
227     if(!readAny())return false;
228
229     char* store[256];
230     value=strtod(ptr2,store);
231     if(strlen(*store)>0) return false; // value is not a number
232
233     return true;
234 };
```

4.8.2.18 bool tpcISGGridSender::readVal (float & value)

Definition at line 214 of file tpcISGGridSender.cc.

```
214                                     {
215
216     if(!readAny()) return false;
217
218     char* store[256];
219     value=(float)strtod(ptr2,store);
220     if(strlen(*store)>0) return false; // value is not a number
221
222     return true;
223 };
```

4.8.2.19 bool tpcISGGridSender::readVal (char *& value)

Definition at line 207 of file tpcISGGridSender.cc.

```

207                                     {
208
209   if(!readAny()) return false;
210   strcpy(value,ptr2);
211   return true;
212 }
```

4.8.2.20 bool tpcISGGridSender::updateDb (const char *fileName) [virtual]

Definition at line 127 of file tpcISGGridSender.cc.

```

127                                     {
128 #define __METHOD__ "updateDb(filename)"
129
130   if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
131
132   int* elements;
133   tpcISGGrid* vals;
134   int numRows = 0;
135
136   if(writeRequired()){
137
138     numRows=NUM_DB_ROWS;
139     elements=elementList;
140     vals = tempVals;
141
142   } else {
143
144     for(int i=0; i<NUM_DB_ROWS; i++){
145       if(hasChanged(i)){
146         updateElements[numRows]=elementList[i];
147         updateVals[numRows] = tempVals[i];
148         previousVals[i]=tempVals[i];
149         numRows++;
150       }
151     }
152
153     elements = updateElements;
154     vals      = updateVals;
155   }
156
157   if(numRows==0) return sendMess(" No update required for",mbaseName,dbMDebug,__LINE__,__CLASS__,__METHOD__);
158
159   //char mess[256];
160   ostringstream sn;
161   sn<<"Will Update "<<numRows<<" of "<<NUM_DB_ROWS<<" rows ";
162   sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
163
164   StDbTable* dbTable=node->findTable("tpcISGGrid");
165   dbTable->SetTable((char*)vals, numRows, elements);
166   mgr->setStoreTime(writeTime);
167
168   if(!mgr->storeDbTable(dbTable)) {
169     addBackLog(writeTime);
170     return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
171   }
172
173   if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
174 }
```

```
175     return true;
176 #undef __METHOD__
177 }
```

4.8.3 Member Data Documentation

4.8.3.1 int [tpcISGGridSender::elementList](#)[NUM_DB_ROWS] [protected]

Definition at line 24 of file tpcISGGridSender.hh.

4.8.3.2 char [tpcISGGridSender::mline](#)[256] [protected]

Definition at line 29 of file tpcISGGridSender.hh.

4.8.3.3 bool [tpcISGGridSender::mreadStatus](#) [protected]

Definition at line 28 of file tpcISGGridSender.hh.

4.8.3.4 tpcISGGrid [tpcISGGridSender::previousVals](#)[NUM_DB_ROWS] [protected]

Definition at line 22 of file tpcISGGridSender.hh.

4.8.3.5 char* [tpcISGGridSender::ptr1](#) [protected]

Definition at line 31 of file tpcISGGridSender.hh.

4.8.3.6 char * [tpcISGGridSender::ptr2](#) [protected]

Definition at line 31 of file tpcISGGridSender.hh.

4.8.3.7 tpcISGGrid [tpcISGGridSender::tempVals](#)[NUM_DB_ROWS] [protected]

Definition at line 23 of file tpcISGGridSender.hh.

4.8.3.8 char [tpcISGGridSender::tmpline](#)[256] [protected]

Definition at line 30 of file tpcISGGridSender.hh.

4.8.3.9 int [tpcISGGridSender::updateElements](#)[NUM_DB_ROWS] [protected]

Definition at line 26 of file tpcISGGridSender.hh.

4.8.3.10 tpcISGGrid [tpcISGGridSender::updateVals](#)[NUM_DB_ROWS] [protected]

Definition at line 25 of file tpcISGGridSender.hh.

4.8.3.11 float [tpcISGGridSender::vdriftLimit](#) [protected]

dito

Definition at line 35 of file tpcISGGridSender.hh.

The documentation for this class was generated from the following files:

- [tpcISGGridSender.hh](#)
- [tpcISGGridSender.cc](#)
- [tpcISGGridSender_i.cc](#)

4.9 tpcOSANodeSender Class Reference

```
#include <tpcOSANodeSender.hh>
```

Public Member Functions

- [tpcOSANodeSender](#) (const char *localDir)
- virtual [~tpcOSANodeSender](#) ()
- virtual void [initTable](#) ()
- virtual void [initTags](#) ()
- virtual void [initDataBase](#) ()
- virtual bool [loadUserControls](#) (const char *name, const char *value)
- virtual void [initQuery](#) ()
- virtual bool [queryData](#) ()
- virtual bool [readData](#) (const char *fileName)
- virtual bool [updateDb](#) (const char *fileName)
- virtual bool [readData](#) (ifstream &from)
- virtual bool [hasChanged](#) (int rowNumber)
- char * [readAny](#) ()
- bool [readVal](#) (char *&value)
- bool [readVal](#) (float &value)
- bool [readVal](#) (double &value)
- bool [readVal](#) (short &value)
- bool [readVal](#) (int &value)
- bool [readVal](#) (long &value)
- bool [readVal](#) (long long &value)
- bool [nextLine](#) (ifstream &from)
- void [readError](#) (int l, char *c, char *m)

Protected Attributes

- tpcOSANode [previousVals](#) [NUM_DB_ROWS]
- tpcOSANode [tempVals](#) [NUM_DB_ROWS]
- int [elementList](#) [NUM_DB_ROWS]
- tpcOSANode [updateVals](#) [NUM_DB_ROWS]
- int [updateElements](#) [NUM_DB_ROWS]
- bool [mreadStatus](#)
- char [mline](#) [256]
- char [tmpline](#) [256]
- char * [ptr1](#)
- char * [ptr2](#)
- float [vdriftLimit](#)
ditto
- float [cdriftLimit](#)

4.9.1 Constructor & Destructor Documentation

4.9.1.1 tpcOSANodeSender::tpcOSANodeSender (const char * *localDir*)

Definition at line 19 of file tpcOSANodeSender.cc.

```

19                                     {
20
21     initTags();
22     if(localDir) cd(localDir);// note this ignores the sub dir tag
23     init("tpcOSANode"); // setup the file I/O
24     initDataBase();        // database connections
25     initTable();           // table definitions
26
27 }
```

4.9.1.2 virtual tpcOSANodeSender::~tpcOSANodeSender () [inline, virtual]

Definition at line 42 of file tpcOSANodeSender.hh.

```
42 {};
```

4.9.2 Member Function Documentation

4.9.2.1 bool tpcOSANodeSender::hasChanged (int *rowNumber*) [virtual]

Definition at line 144 of file tpcOSANodeSender_i.cc.

```

144                                     {
145
146     tpcOSANode* pre=&previousVals[rowNumber];
147     tpcOSANode* cur=&tempVals[rowNumber];
148
149     if(fabs(pre->mVolt1-cur->mVolt1)>=vdriftLimit) return true;
150     if(fabs(pre->mVolt2-cur->mVolt2)>=vdriftLimit) return true;
151     if(fabs(pre->mVolt3-cur->mVolt3)>=vdriftLimit) return true;
152     if(fabs(pre->mVolt4-cur->mVolt4)>=vdriftLimit) return true;
153     if(fabs(pre->mCurrent1-cur->mCurrent1)>=cdriftLimit) return true;
154     if(fabs(pre->mCurrent2-cur->mCurrent2)>=cdriftLimit) return true;
155     if(fabs(pre->mCurrent3-cur->mCurrent3)>=cdriftLimit) return true;
156     if(fabs(pre->mCurrent4-cur->mCurrent4)>=cdriftLimit) return true;
157
158     /* example ... note -> change to any element requires db-update
159     * and thus returns true immediately
160     *
161     *if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
162     *if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
163     *
164     * ....
165     */
166
167     return false;
168 }
```

4.9.2.2 void tpcOSANodeSender::initDataBase () [virtual]

Definition at line 75 of file tpcOSANodeSender.cc.

```

75                                     {
76 #define __METHOD__ "initDataBase()"
77
78 /* More than an example... swap user & dbTrg as per subsystem*/
79 mgr->setUser("stardb","");
80 StDbType   dbT = dbConditions;
81 StDbDomain dbD = dbTpc;
82
83 if( !( node = mgr->initConfig(dbT,dbD) ) )
84     sendMess("Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal,__LINE__,__CLASS__,__METHOD__);
85
86 #undef __METHOD__
87 }

```

4.9.2.3 void tpcOSANodeSender::initQuery () [virtual]

Definition at line 48 of file tpcOSANodeSender_i.cc.

```

48                                     {
49 #define __METHOD__ "initQuery()"
50
51     ofstream to(queryFile);
52
53     if(!to.is_open()){
54         sendMess("Open Failed ",queryFile,dbMFatal,__LINE__,__CLASS__,__METHOD__);
55         return;
56     }
57
58     const char *bar[8] = {"E","F","G","H","I","J","K","L"};
59     int i=0; // outer sector
60
61     for(int j = 0; j < 12; j++){ // card number
62         for(int k = 0; k < 8; k++){ //channel number
63             to<<"tpchv:MC_"<<i<<"_"<<j<<"_"<<k<<endl; //measured current
64             to<<"tpchv:SUB_RD_MV_"<<i<<"_"<<j<<"_"<<bar[k]<<endl; // measured volt
65             to<<"tpchv:SUB_RD_ST_"<<i<<"_"<<j<<"_"<<bar[k]<<endl; //trip status
66             to<<"tpchv:SUB_RD_CE_"<<i<<"_"<<j<<"_"<<bar[k]<<endl; //enabled
67             to<<"tpchv:SUB_HV_"<<i<<"_"<<j<<"_"<<bar[k]<<endl; //on
68         }
69     }
70
71
72 /* example
73 *     for(int i=0;i<16;i++){
74 *         to<<"TRGhv:SUB_RD_V_1:"<<i<<".E"<<endl;
75 *         to<<"TRGhv:SUB_RD_V_1:"<<i<<".F"<<endl;
76 *         ....
77 *     }
78 *
79 */
80
81     to.close();
82
83 #undef __METHOD__
84 }

```

4.9.2.4 void tpcOSANodeSender::initTable () [virtual]

Definition at line 30 of file tpcOSANodeSender.cc.

```

30                                     {

```

```

31 #define __METHOD__ "initTable()"
32
33     StDbTable* table=0;
34     if(!(table=node->addDbTable("tpcOSANode")))
35         sendMess("Could not find table=tpcOSANode",dbmFatal,__LINE__,__CLASS__,__METHOD__);
36
37     memset(tempVals,0,NUM_DB_ROWS*sizeof(tpcOSANode));
38     memset(previousVals,0,NUM_DB_ROWS*sizeof(tpcOSANode));
39
40     int nrows;
41     int* elist = table->getElementID(nrows);
42     if(nrows!=NUM_DB_ROWS){
43         //char mess[256];
44         ostringstream ms;
45         ms<<"Db rows("<<nrows<<") != compiled("<<NUM_DB_ROWS<<")";
46         sendMess((ms.str()).c_str(),dbmFatal,__LINE__,__CLASS__,__METHOD__);
47     }
48     memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
49
50     unsigned int timestamp=time(NULL);
51     mgr->setRequestTime(timestamp);
52     if(mgr->fetchDbTable(table)){
53         tpcOSANode* thv = (tpcOSANode*)table->GetTable();
54         memcpy(previousVals,thv,nrows*sizeof(tpcOSANode));
55     }
56
57 #undef __METHOD__
58 };

```

4.9.2.5 void tpcOSANodeSender::initTags () [virtual]

Definition at line 66 of file tpcOSANodeSender.cc.

```

66     {
67     /* more than an example -> swap "trg" to your subsys & add to email list*/
68     setEmailTo("porter@bnl.gov");
69     setDomainName("tpc");
70
71 }

```

4.9.2.6 bool tpcOSANodeSender::loadUserControls (const char * name, const char * value) [virtual]

Definition at line 20 of file tpcOSANodeSender_i.cc.

```

20     {
21 #define __METHOD__ "loadUserControls(name,value)"
22
23 /* more than an example ... swap driftLimit to yours
24 * and duplicate this structure for each selection criteria
25 if(strstr(name,"driftLimit")){
26     driftLimit=atof(value);
27     sendMess("driftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
28     return true;
29 }
30 */
31 if(strstr(name,"vdriftLimit")){
32     vdriftLimit=atof(value);
33     sendMess("vdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
34     return true;

```

```

35 }
36 if(strstr(name,"cdriftLimit")){
37     cdriftLimit=atof(value);
38     sendMess("cdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
39     return true;
40 }
41
42 return false;
43 #undef __METHOD__
44 }

```

4.9.2.7 bool tpcOSANodeSender::nextLine (ifstream &from) [inline]

Definition at line 75 of file tpcOSANodeSender.hh.

```

75                                     {
76     if(!from.getline(mline,255))return false;
77     return true;
78 }

```

4.9.2.8 bool tpcOSANodeSender::queryData () [virtual]

Definition at line 91 of file tpcOSANodeSender.cc.

```

91                                     {
92 #define __METHOD__ "queryData()"
93
94 /*
95  * MORE THAN AN EXAMPLE...
96  * IF Standard SC-Query via "caGet" then,
97  * no need to change this method AT ALL
98  *
99  */
100
101     writeTime = (unsigned int)time(NULL);          //for database write time
102
103     //char systemCmd[1024];
104     ostringstream scmd;
105     scmd<<"caGet "<<queryFile<<" "<<dataFile;
106
107     if(system((scmd.str()).c_str()))
108         return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
109
110     return true;
111 #undef __METHOD__
112 };

```

4.9.2.9 char * tpcOSANodeSender::readAny ()

Definition at line 197 of file tpcOSANodeSender.cc.

```

197                                     {
198
199     strcpy(tmpline,mline);
200     ptr1=tmpline;
201     ptr2=strtok(ptr1," ");
202     if(!ptr2) return ptr2;

```

```

203 ptr2=strtok(NULL, " ");
204 return ptr2;
205 }

```

4.9.2.10 bool tpcOSANodeSender::readData (ifstream & from) [virtual]

Definition at line 89 of file tpcOSANodeSender_i.cc.

```

89                                     {
90 #define __METHOD__ "readData(ifstream)"
91
92 mreadStatus=true;
93 memset(tempVals,0,NUM_DB_ROWS*sizeof(tpcOSANode));
94
95
96 //int i=0;
97 char* c=__CLASS__;
98 char* m=__METHOD__;
99
100
101 for(int i=0;i<24;i++){
102
103     if(!nextLine(from) || !readVal(tempVals[i].mCurrent4)) readError(__LINE__,c,m);
104     if(!nextLine(from) || !readVal(tempVals[i].mVolt4)) readError(__LINE__,c,m);
105     if(!nextLine(from) || !readVal(tempVals[i].trip4)) readError(__LINE__,c,m);
106     if(!nextLine(from) || !readVal(tempVals[i].enable4)) readError(__LINE__,c,m);
107     if(!nextLine(from) || !readVal(tempVals[i].channelOn4)) readError(__LINE__,c,m);
108
109     if(!nextLine(from) || !readVal(tempVals[i].mCurrent3)) readError(__LINE__,c,m);
110     if(!nextLine(from) || !readVal(tempVals[i].mVolt3)) readError(__LINE__,c,m);
111     if(!nextLine(from) || !readVal(tempVals[i].trip3)) readError(__LINE__,c,m);
112     if(!nextLine(from) || !readVal(tempVals[i].enable3)) readError(__LINE__,c,m);
113     if(!nextLine(from) || !readVal(tempVals[i].channelOn3)) readError(__LINE__,c,m);
114
115     if(!nextLine(from) || !readVal(tempVals[i].mCurrent2)) readError(__LINE__,c,m);
116     if(!nextLine(from) || !readVal(tempVals[i].mVolt2)) readError(__LINE__,c,m);
117     if(!nextLine(from) || !readVal(tempVals[i].trip2)) readError(__LINE__,c,m);
118     if(!nextLine(from) || !readVal(tempVals[i].enable2)) readError(__LINE__,c,m);
119     if(!nextLine(from) || !readVal(tempVals[i].channelOn2)) readError(__LINE__,c,m);
120
121     if(!nextLine(from) || !readVal(tempVals[i].mCurrent1)) readError(__LINE__,c,m);
122     if(!nextLine(from) || !readVal(tempVals[i].mVolt1)) readError(__LINE__,c,m);
123     if(!nextLine(from) || !readVal(tempVals[i].trip1)) readError(__LINE__,c,m);
124     if(!nextLine(from) || !readVal(tempVals[i].enable1)) readError(__LINE__,c,m);
125     if(!nextLine(from) || !readVal(tempVals[i].channelOn1)) readError(__LINE__,c,m);
126 }
127
128 /* example format
129 * for(int i=0;i<NUM_DB_ROWS;i++){
130 * if(!nextLine(from) || !readVal(tempVals[i].blah)) readError(l,c,m);
131 *
132 *     ....
133 *
134 * }
135 */
136
137 from.close();
138 return true;
139 #undef __METHOD__
140 }

```

4.9.2.11 bool tpcOSANodeSender::readData (const char **fileName*) [virtual]

Definition at line 116 of file tpcOSANodeSender.cc.

```
116                                     {
117 #define __METHOD__ "readData(fileName)"
118
119     ifstream from(fileName);
120     if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
121
122     return readData(from); // user implemented file read
123 #undef __METHOD__
124 }
```

4.9.2.12 void tpcOSANodeSender::readError (int *l*, char * *c*, char * *m*) [inline]

Definition at line 80 of file tpcOSANodeSender.hh.

```
80                                     {
81     mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
82 }
```

4.9.2.13 bool tpcOSANodeSender::readVal (long long & *value*)

Definition at line 269 of file tpcOSANodeSender.cc.

```
269                                     {
270
271     if(!readAny())return false;
272     char* store[256];
273     value=strtoll(ptr2,store,10);
274     if(strlen(*store)>0) return false; // value is not a number
275
276     return true;
277 };
```

4.9.2.14 bool tpcOSANodeSender::readVal (long & *value*)

Definition at line 258 of file tpcOSANodeSender.cc.

```
258                                     {
259
260     if(!readAny())return false;
261
262     char* store[256];
263     value=strtoll(ptr2,store,10);
264     if(strlen(*store)>0) return false; // value is not a number
265
266     return true;
267 };
```

4.9.2.15 bool tpcOSANodeSender::readVal (int & value)

Definition at line 247 of file tpcOSANodeSender.cc.

```
247                                     {
248
249     if(!readAny()) return false;
250
251     char* store[256];
252     value=(int)strtol(ptr2,store,10);
253     if(strlen(*store)>0) return false; // value is not a number
254
255     return true;
256 };
```

4.9.2.16 bool tpcOSANodeSender::readVal (short & value)

Definition at line 236 of file tpcOSANodeSender.cc.

```
236                                     {
237
238     if(!readAny()) return false;
239
240     char* store[256];
241     value=(short)strtol(ptr2,store,10);
242     if(strlen(*store)>0) return false; // value is not a number
243
244     return true;
245 };
```

4.9.2.17 bool tpcOSANodeSender::readVal (double & value)

Definition at line 225 of file tpcOSANodeSender.cc.

```
225                                     {
226
227     if(!readAny())return false;
228
229     char* store[256];
230     value=strtod(ptr2,store);
231     if(strlen(*store)>0) return false; // value is not a number
232
233     return true;
234 };
```

4.9.2.18 bool tpcOSANodeSender::readVal (float & value)

Definition at line 214 of file tpcOSANodeSender.cc.

```
214                                     {
215
216     if(!readAny()) return false;
217
218     char* store[256];
219     value=(float)strtod(ptr2,store);
220     if(strlen(*store)>0) return false; // value is not a number
221
222     return true;
223 };
```

4.9.2.19 bool tpcOSANodeSender::readVal (char *& value)

Definition at line 207 of file tpcOSANodeSender.cc.

```

207                                     {
208
209     if(!readAny()) return false;
210     strcpy(value,ptr2);
211     return true;
212 }
```

4.9.2.20 bool tpcOSANodeSender::updateDb (const char *fileName) [virtual]

Definition at line 127 of file tpcOSANodeSender.cc.

```

127                                     {
128 #define __METHOD__ "updateDb(filename)"
129
130     if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
131
132     int* elements;
133     tpcOSANode* vals;
134     int numRows = 0;
135
136     if(writeRequired()){
137
138         numRows=NUM_DB_ROWS;
139         elements=elementList;
140         vals = tempVals;
141
142     } else {
143
144         for(int i=0; i<NUM_DB_ROWS; i++){
145             if(hasChanged(i)){
146                 updateElements[numRows]=elementList[i];
147                 updateVals[numRows] = tempVals[i];
148                 previousVals[i]=tempVals[i];
149                 numRows++;
150             }
151         }
152
153         elements = updateElements;
154         vals      = updateVals;
155     }
156
157     if(numRows==0) return sendMess(" No update required for",mbaseName,dbMDebug,__LINE__,__CLASS__,__METHOD__);
158
159     //char mess[256];
160     ostreamstream sn;
161     sn<<"Will Update "<<numRows<<" of "<<NUM_DB_ROWS<<" rows ";
162     sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
163
164     StDbTable* dbTable=node->findTable("tpcOSANode");
165     dbTable->SetTable((char*)vals, numRows, elements);
166     mgr->setStoreTime(writeTime);
167
168     if(!mgr->storeDbTable(dbTable)) {
169         addBackLog(writeTime);
170         return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
171     }
172
173     if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
174 }
```

```
175     return true;
176     #undef __METHOD__
177 }
```

4.9.3 Member Data Documentation

4.9.3.1 float [tpcOSANodeSender::cdriftLimit](#) [protected]

Definition at line 36 of file tpcOSANodeSender.hh.

4.9.3.2 int [tpcOSANodeSender::elementList](#)[NUM_DB_ROWS] [protected]

Definition at line 24 of file tpcOSANodeSender.hh.

4.9.3.3 char [tpcOSANodeSender::mline](#)[256] [protected]

Definition at line 29 of file tpcOSANodeSender.hh.

4.9.3.4 bool [tpcOSANodeSender::mreadStatus](#) [protected]

Definition at line 28 of file tpcOSANodeSender.hh.

4.9.3.5 tpcOSANode [tpcOSANodeSender::previousVals](#)[NUM_DB_ROWS] [protected]

Definition at line 22 of file tpcOSANodeSender.hh.

4.9.3.6 char* [tpcOSANodeSender::ptr1](#) [protected]

Definition at line 31 of file tpcOSANodeSender.hh.

4.9.3.7 char * [tpcOSANodeSender::ptr2](#) [protected]

Definition at line 31 of file tpcOSANodeSender.hh.

4.9.3.8 tpcOSANode [tpcOSANodeSender::tempVals](#)[NUM_DB_ROWS] [protected]

Definition at line 23 of file tpcOSANodeSender.hh.

4.9.3.9 char [tpcOSANodeSender::tmpline](#)[256] [protected]

Definition at line 30 of file tpcOSANodeSender.hh.

4.9.3.10 int [tpcOSANodeSender::updateElements](#)[NUM_DB_ROWS] [protected]

Definition at line 26 of file tpcOSANodeSender.hh.

4.9.3.11 `tpcOSAnode` `tpcOSAnodeSender::updateVals[NUM_DB_ROWS]` [`protected`]

Definition at line 25 of file `tpcOSAnodeSender.hh`.

4.9.3.12 `float` `tpcOSAnodeSender::vdriftLimit` [`protected`]

dito

Definition at line 35 of file `tpcOSAnodeSender.hh`.

The documentation for this class was generated from the following files:

- [tpcOSAnodeSender.hh](#)
- [tpcOSAnodeSender.cc](#)
- [tpcOSAnodeSender_i.cc](#)

4.10 tpcOSGGridSender Class Reference

```
#include <tpcOSGGridSender.h>
```

Public Member Functions

- [tpcOSGGridSender](#) (const char *localDir)
- virtual [~tpcOSGGridSender](#) ()
- virtual void [initTable](#) ()
- virtual void [initTags](#) ()
- virtual void [initDataBase](#) ()
- virtual bool [loadUserControls](#) (const char *name, const char *value)
- virtual void [initQuery](#) ()
- virtual bool [queryData](#) ()
- virtual bool [readData](#) (const char *fileName)
- virtual bool [updateDb](#) (const char *fileName)
- virtual bool [readData](#) (ifstream &from)
- virtual bool [hasChanged](#) (int rowNumber)
- char * [readAny](#) ()
- bool [readVal](#) (char *&value)
- bool [readVal](#) (float &value)
- bool [readVal](#) (double &value)
- bool [readVal](#) (short &value)
- bool [readVal](#) (int &value)
- bool [readVal](#) (long &value)
- bool [readVal](#) (long long &value)
- bool [nextLine](#) (ifstream &from)
- void [readError](#) (int l, char *c, char *m)

Protected Attributes

- tpcOSGGrid [previousVals](#) [NUM_DB_ROWS]
- tpcOSGGrid [tempVals](#) [NUM_DB_ROWS]
- int [elementList](#) [NUM_DB_ROWS]
- tpcOSGGrid [updateVals](#) [NUM_DB_ROWS]
- int [updateElements](#) [NUM_DB_ROWS]
- bool [mreadStatus](#)
- char [mline](#) [256]
- char [tmpline](#) [256]
- char * [ptr1](#)
- char * [ptr2](#)
- float [vdriftLimit](#)

ditto

4.10.1 Constructor & Destructor Documentation

4.10.1.1 tpcOSGGridSender::tpcOSGGridSender (const char * *localDir*)

Definition at line 19 of file tpcOSGGridSender.cc.

```

19                                     {
20
21     initTags();
22     if(localDir) cd(localDir);// note this ignores the sub dir tag
23     init("tpcOSGGrid"); // setup the file I/O
24     initDataBase();      // database connections
25     initTable();         // table definitions
26
27 }
```

4.10.1.2 virtual tpcOSGGridSender::~tpcOSGGridSender () [inline, virtual]

Definition at line 41 of file tpcOSGGridSender.hh.

```
41 {};
```

4.10.2 Member Function Documentation

4.10.2.1 bool tpcOSGGridSender::hasChanged (int *rowNumber*) [virtual]

Definition at line 149 of file tpcOSGGridSender_i.cc.

```

149                                     {
150
151     tpcOSGGrid* pre=&previousVals[rowNumber];
152     tpcOSGGrid* cur=&tempVals[rowNumber];
153
154     if(fabs(pre->refVolt-cur->refVolt)>=vdriftLimit) return true;
155     if(fabs(pre->hiVolt-cur->hiVolt)>=vdriftLimit) return true;
156     if(fabs(pre->loVolt-cur->loVolt)>=vdriftLimit) return true;
157
158     /* example ... note -> change to any element requires db-update
159      * and thus returns true immediately
160      *
161      *if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
162      *if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
163      *
164      * ....
165      */
166
167     return false;
168 }
```

4.10.2.2 void tpcOSGGridSender::initDataBase () [virtual]

Definition at line 75 of file tpcOSGGridSender.cc.

```

75                                     {
76 #define __METHOD__ "initDataBase()"
```

```

77
78  /* More than an example... swap user & dbTrg as per subsystem*/
79  mgr->setUser("stardb","");
80  StDbType   dbT = dbConditions;
81  StDbDomain dbD = dbTpc;
82
83  if( !( node = mgr->initConfig(dbT,dbD) ) )
84      sendMess("Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal, __LINE__, __CLASS__, __METHOD__);
85
86 #undef __METHOD__
87 }

```

4.10.2.3 void tpcOSGGridSender::initQuery () [virtual]

Definition at line 38 of file tpcOSGGridSender_i.cc.

```

38  {
39 #define __METHOD__ "initQuery()"
40
41  ofstream to(queryFile);
42
43  if(!to.is_open()){
44      sendMess("Open Failed ",queryFile,dbMFatal, __LINE__, __CLASS__, __METHOD__);
45      return;
46  }
47
48  int i;
49  for(i = 0; i <12; i++){
50      to<<"GG_A_In_A_"<<i+1<<endl;
51      to<<"GG_A_In_B_"<<i+1<<endl;
52      to<<"GG_A_In_C_"<<i+1<<endl;
53      to<<"GG_D_Out_D_"<<i+1<<endl;
54      to<<"GG_D_Out_B_"<<i+1<<endl;
55      to<<"GG_D_Out_C_"<<(i/4 + 1)*4<<endl;
56      to<<"GG_D_Out_A_"<<(i/4 + 1)*4<<endl;
57
58  }
59  for(i = 24; i <36; i++){
60      to<<"GG_A_In_A_"<<i+1<<endl;
61      to<<"GG_A_In_B_"<<i+1<<endl;
62      to<<"GG_A_In_C_"<<i+1<<endl;
63      to<<"GG_D_Out_D_"<<i+1<<endl;
64      to<<"GG_D_Out_B_"<<i+1<<endl;
65      to<<"GG_D_Out_C_"<<(i/4 + 1)*4<<endl;
66      to<<"GG_D_Out_A_"<<(i/4 + 1)*4<<endl;
67  }
68  /* example
69  *   for(int i=0;i<16;i++){
70  *       to<<"TRGhv:SUB_RD_V_1:"<<i<<".E"<<endl;
71  *       to<<"TRGhv:SUB_RD_V_1:"<<i<<".F"<<endl;
72  *       ....
73  *   }
74  * */
75
76
77  to.close();
78
79 #undef __METHOD__
80 }

```

4.10.2.4 void tpcOSGGridSender::initTable () [virtual]

Definition at line 30 of file tpcOSGGridSender.cc.

```

30
31 #define __METHOD__ "initTable()"
32
33     StDbTable* table=0;
34     if(!(table=node->addDbTable("tpcOSGGrid")))
35         sendMess("Could not find table=tpcOSGGrid",dbmFatal,__LINE__,__CLASS__,__METHOD__);
36
37     memset(tempVals,0,NUM_DB_ROWS*sizeof(tpcOSGGrid));
38     memset(previousVals,0,NUM_DB_ROWS*sizeof(tpcOSGGrid));
39
40     int nrows;
41     int* elist = table->getElementID(nrows);
42     if(nrows!=NUM_DB_ROWS){
43         //char mess[256];
44         ostringstream ms;
45         ms<<"Db rows("<<nrows<<") != compiled("<<NUM_DB_ROWS<<")";
46         sendMess((ms.str()).c_str(),dbmFatal,__LINE__,__CLASS__,__METHOD__);
47     }
48     memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
49
50     unsigned int timestamp=time(NULL);
51     mgr->setRequestTime(timestamp);
52     if(mgr->fetchDbTable(table)){
53         tpcOSGGrid* thv = (tpcOSGGrid*)table->GetTable();
54         memcpy(previousVals,thv,nrows*sizeof(tpcOSGGrid));
55     }
56
57 #undef __METHOD__
58 };

```

4.10.2.5 void tpcOSGGridSender::inifTags () [virtual]

Definition at line 66 of file tpcOSGGridSender.cc.

```

66
67     /* more than an example -> swap "trg" to your subsys & add to email list*/
68     setEmailTo("porter@bnl.gov");
69     setDomainName("tpc");
70
71 }

```

4.10.2.6 bool tpcOSGGridSender::loadUserControls (const char * name, const char * value) [virtual]

Definition at line 20 of file tpcOSGGridSender_i.cc.

```

20
21 #define __METHOD__ "loadUserControls(name,value)"
22
23 /* more than an example ... swap driftLimit to yours
24 * and duplicate this structure for each selection criteria */
25 if(strstr(name,"vdriftLimit")){
26     vdriftLimit=atof(value);
27     sendMess("vdriftLimit set=",value,dbmDebug,__LINE__,__CLASS__,__METHOD__);
28     return true;
29 }
30
31
32 return false;
33 #undef __METHOD__
34 }

```

4.10.2.7 bool tpcOSGGridSender::nextLine (ifstream & from) [inline]

Definition at line 74 of file tpcOSGGridSender.hh.

```

74                                     {
75   if(!from.getline(mline,255))return false;
76   return true;
77 }
```

4.10.2.8 bool tpcOSGGridSender::queryData () [virtual]

Definition at line 91 of file tpcOSGGridSender.cc.

```

91                                     {
92 #define __METHOD__ "queryData()"
93
94 /*
95  * MORE THAN AN EXAMPLE...
96  * IF Standard SC-Query via "caGet" then,
97  * no need to change this method AT ALL
98  *
99  */
100
101 writeTime = (unsigned int)time(NULL); //for database write time
102
103 //char systemCmd[1024];
104 ostringstream scmd;
105 scmd<<"caGet "<<queryFile<<" "<<dataFile;
106
107 if(system((scmd.str()).c_str()))
108   return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
109
110 return true;
111 #undef __METHOD__
112 };
```

4.10.2.9 char * tpcOSGGridSender::readAny ()

Definition at line 197 of file tpcOSGGridSender.cc.

```

197                                     {
198
199   strcpy(tmpline,mline);
200   ptr1=tmpline;
201   ptr2=strtok(ptr1," ");
202   if(!ptr2) return ptr2;
203   ptr2=strtok(NULL," ");
204   return ptr2;
205 }
```

4.10.2.10 bool tpcOSGGridSender::readData (ifstream & from) [virtual]

Definition at line 85 of file tpcOSGGridSender_i.cc.

```

85                                     {
86 #define __METHOD__ "readData(ifstream)"
```

```

87
88 mreadStatus=true;
89 memset(tempVals,0,NUM_DB_ROWS*sizeof(tpcOSGGrid));
90
91
92 //int i=0;
93 char* c=__CLASS__;
94 char* m=__METHOD__;
95
96 for(int i=0;i<24;i++){
97     if(!nextLine(from) || !readVal(tempVals[i].refVolt)) readError(__LINE__,c,m);
98     if(!nextLine(from) || !readVal(tempVals[i].hiVolt)) readError(__LINE__,c,m);
99     if(!nextLine(from) || !readVal(tempVals[i].loVolt)) readError(__LINE__,c,m);
100 //MPD changes
101     char* tmpval0;
102     if(!nextLine(from)){
103         tmpval0=tempVals[i].hvEnable;
104         if(!readVal(tmpval0))readError(__LINE__,c,m);
105     }
106
107 //     if(!nextLine(from) || !readVal(tempVals[i].hvEnable)) readError(__LINE__,c,m);
108 char* tmpval1;
109
110     if(!nextLine(from)){
111         tmpval1=tempVals[i].psEnable;
112         if(!readVal(tmpval1))readError(__LINE__,c,m);
113     }
114
115 //     if(!nextLine(from) || !readVal(tempVals[i].psEnable)) readError(__LINE__,c,m);
116 char* tmpval2;
117
118     if(!nextLine(from)){
119         tmpval2=tempVals[i].GateEnable;
120         if(!readVal(tmpval2))readError(__LINE__,c,m);
121     }
122
123 //     if(!nextLine(from) || !readVal(tempVals[i].GateEnable)) readError(__LINE__,c,m);
124 char* tmpval3;
125
126     if(!nextLine(from)){
127         tmpval3=tempVals[i].trigEnable;
128         if(!readVal(tmpval3))readError(__LINE__,c,m);
129     }
130
131 //     if(!nextLine(from) || !readVal(tempVals[i].trigEnable)) readError(__LINE__,c,m);
132 }
133 /* example format
134 * for(int i=0;i<NUM_DB_ROWS;i++){
135 * if(!nextLine(from) || !readVal(tempVals[i].blah)) readError(l,c,m);
136 *
137 *     ....
138 *
139 * }
140 */
141
142 from.close();
143 return true;
144 #undef __METHOD__
145 }

```

4.10.2.11 bool tpcOSGGridSender::readData (const char * *fileName*) [virtual]

Definition at line 116 of file tpcOSGGridSender.cc.

```

116                                     {

```

```

117 #define __METHOD__ "readData(fileName)"
118
119 ifstream from(fileName);
120 if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
121
122 return readData(from); // user implemented file read
123 #undef __METHOD__
124 }

```

4.10.2.12 void tpcOSGGridSender::readError (int *l*, char * *c*, char * *m*) [inline]

Definition at line 79 of file tpcOSGGridSender.hh.

```

79
80 mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
81 }

```

4.10.2.13 bool tpcOSGGridSender::readVal (long long & *value*)

Definition at line 269 of file tpcOSGGridSender.cc.

```

269
270
271 if(!readAny())return false;
272 char* store[256];
273 value=strtoll(ptr2,store,10);
274 if(strlen(*store)>0) return false; // value is not a number
275
276 return true;
277 };

```

4.10.2.14 bool tpcOSGGridSender::readVal (long & *value*)

Definition at line 258 of file tpcOSGGridSender.cc.

```

258
259
260 if(!readAny())return false;
261
262 char* store[256];
263 value=strtol(ptr2,store,10);
264 if(strlen(*store)>0) return false; // value is not a number
265
266 return true;
267 };

```

4.10.2.15 bool tpcOSGGridSender::readVal (int & *value*)

Definition at line 247 of file tpcOSGGridSender.cc.

```

247
248
249 if(!readAny()) return false;
250

```

```
251 char* store[256];
252 value=(int)strtol(ptr2,store,10);
253 if(strlen(*store)>0) return false; // value is not a number
254
255 return true;
256 };
```

4.10.2.16 bool tpcOSGGridSender::readVal (short & value)

Definition at line 236 of file tpcOSGGridSender.cc.

```
236                                     {
237
238     if(!readAny()) return false;
239
240     char* store[256];
241     value=(short)strtol(ptr2,store,10);
242     if(strlen(*store)>0) return false; // value is not a number
243
244     return true;
245 };
```

4.10.2.17 bool tpcOSGGridSender::readVal (double & value)

Definition at line 225 of file tpcOSGGridSender.cc.

```
225                                     {
226
227     if(!readAny())return false;
228
229     char* store[256];
230     value=strtod(ptr2,store);
231     if(strlen(*store)>0) return false; // value is not a number
232
233     return true;
234 };
```

4.10.2.18 bool tpcOSGGridSender::readVal (float & value)

Definition at line 214 of file tpcOSGGridSender.cc.

```
214                                     {
215
216     if(!readAny()) return false;
217
218     char* store[256];
219     value=(float)strtod(ptr2,store);
220     if(strlen(*store)>0) return false; // value is not a number
221
222     return true;
223 };
```

4.10.2.19 bool tpcOSGGridSender::readVal (char *& value)

Definition at line 207 of file tpcOSGGridSender.cc.

```

207                                     {
208
209     if(!readAny()) return false;
210     strcpy(value,ptr2);
211     return true;
212 }
```

4.10.2.20 bool tpcOSGGridSender::updateDb (const char *fileName) [virtual]

Definition at line 127 of file tpcOSGGridSender.cc.

```

127                                     {
128 #define __METHOD__ "updateDb(filename)"
129
130     if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
131
132     int* elements;
133     tpcOSGGrid* vals;
134     int numRows = 0;
135
136     if(writeRequired()){
137
138         numRows=NUM_DB_ROWS;
139         elements=elementList;
140         vals = tempVals;
141
142     } else {
143
144         for(int i=0; i<NUM_DB_ROWS; i++){
145             if(hasChanged(i)){
146                 updateElements[numRows]=elementList[i];
147                 updateVals[numRows] = tempVals[i];
148                 previousVals[i]=tempVals[i];
149                 numRows++;
150             }
151         }
152
153         elements = updateElements;
154         vals      = updateVals;
155     }
156
157     if(numRows==0) return sendMess(" No update required for",mbaseName,dbMDebug,__LINE__,__CLASS__,__METHOD__);
158
159     //char mess[256];
160     ostringstream sn;
161     sn<<"Will Update "<<numRows<<" of "<<NUM_DB_ROWS<<" rows ";
162     sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
163
164     StDbTable* dbTable=node->findTable("tpcOSGGrid");
165     dbTable->SetTable((char*)vals, numRows, elements);
166     mgr->setStoreTime(writeTime);
167
168     if(!mgr->storeDbTable(dbTable)) {
169         addBackLog(writeTime);
170         return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
171     }
172
173     if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
174 }
```

```
175     return true;
176     #undef __METHOD__
177 }
```

4.10.3 Member Data Documentation

4.10.3.1 int [tpcOSGGridSender::elementList](#)[NUM_DB_ROWS] [protected]

Definition at line 24 of file tpcOSGGridSender.hh.

4.10.3.2 char [tpcOSGGridSender::mline](#)[256] [protected]

Definition at line 29 of file tpcOSGGridSender.hh.

4.10.3.3 bool [tpcOSGGridSender::mreadStatus](#) [protected]

Definition at line 28 of file tpcOSGGridSender.hh.

4.10.3.4 tpcOSGGrid [tpcOSGGridSender::previousVals](#)[NUM_DB_ROWS] [protected]

Definition at line 22 of file tpcOSGGridSender.hh.

4.10.3.5 char* [tpcOSGGridSender::ptr1](#) [protected]

Definition at line 31 of file tpcOSGGridSender.hh.

4.10.3.6 char * [tpcOSGGridSender::ptr2](#) [protected]

Definition at line 31 of file tpcOSGGridSender.hh.

4.10.3.7 tpcOSGGrid [tpcOSGGridSender::tempVals](#)[NUM_DB_ROWS] [protected]

Definition at line 23 of file tpcOSGGridSender.hh.

4.10.3.8 char [tpcOSGGridSender::tmpline](#)[256] [protected]

Definition at line 30 of file tpcOSGGridSender.hh.

4.10.3.9 int [tpcOSGGridSender::updateElements](#)[NUM_DB_ROWS] [protected]

Definition at line 26 of file tpcOSGGridSender.hh.

4.10.3.10 tpcOSGGrid [tpcOSGGridSender::updateVals](#)[NUM_DB_ROWS] [protected]

Definition at line 25 of file tpcOSGGridSender.hh.

4.10.3.11 float [tpcOSGGridSender::vdriftLimit](#) [protected]

dito

Definition at line 35 of file tpcOSGGridSender.hh.

The documentation for this class was generated from the following files:

- [tpcOSGGridSender.hh](#)
- [tpcOSGGridSender.cc](#)
- [tpcOSGGridSender_i.cc](#)

Chapter 5

Doxygen_MPD_STAR_ONLINE_DB File Documentation

5.1 tpcAnodeStateDaemon.cc File Reference

```
#include "tpcAnodeStateSender_i.hh"  
#include <unistd.h>
```

Functions

- void [runSender](#) (const char *l`dir`)

5.1.1 Function Documentation

5.1.1.1 void runSender (const char * *l`dir`*)

Definition at line 14 of file tpcAnodeStateDaemon.cc.

```
14         {  
15  
16     CndDbSender* sender = new tpcAnodeStateSender_i(ldir);  
17  
18     sender->initQuery();  
19     for(;;) { //ever...  
20         if(sender->hasBackLog())sender->cleanBackLog();  
21         if(sender->queryData())sender->updateDb();  
22         sleep(sender->sleepTime());  
23     }  
24  
25 };
```

5.2 tpcAnodeStateSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include "tpcAnodeStateSender.hh"
#include "StDbTable.h"
```

Defines

- #define [__CLASS__](#) "tpcAnodeStateSender"
- #define [__METHOD__](#) "initTable()"
- #define [__METHOD__](#) "readData(fileName)"
- #define [__METHOD__](#) "updateDb(filename)"

5.2.1 Define Documentation

5.2.1.1 #define [__CLASS__](#) "tpcAnodeStateSender"

Definition at line 16 of file tpcAnodeStateSender.cc.

5.2.1.2 #define [__METHOD__](#) "updateDb(filename)"

5.2.1.3 #define [__METHOD__](#) "readData(fileName)"

5.2.1.4 #define [__METHOD__](#) "initTable()"

5.3 tpcAnodeStateSender.hh File Reference

```
#include "CndDbSender.hh"  
#include "tpcAnodeState.h"
```

Classes

- class [tpcAnodeStateSender](#)

Defines

- #define [NUM_DB_ROWS](#) 1

5.3.1 Define Documentation

5.3.1.1 #define NUM_DB_ROWS 1

Definition at line 16 of file tpcAnodeStateSender.hh.

5.4 tpcAnodeStateSender_i.cc File Reference

```
#include "tpcAnodeStateSender_i.hh"
```

Defines

- #define `__CLASS__` "tpcAnodeStateSender_i"
- #define `__METHOD__` "initDataBase()"
- #define `__METHOD__` "loadUserControls(name,value)"
- #define `__METHOD__` "queryData()"
- #define `__METHOD__` "readData(ifstream)"

5.4.1 Define Documentation

5.4.1.1 #define `__CLASS__` "tpcAnodeStateSender_i"

Definition at line 11 of file tpcAnodeStateSender_i.cc.

5.4.1.2 #define `__METHOD__` "readData(ifstream)"

5.4.1.3 #define `__METHOD__` "queryData()"

5.4.1.4 #define `__METHOD__` "loadUserControls(name,value)"

5.4.1.5 #define `__METHOD__` "initDataBase()"

5.5 tpcAnodeStateSender_i.hh File Reference

```
#include "tpcAnodeStateSender.hh"
```

Classes

- class [tpcAnodeStateSender_i](#)

5.6 tpcDewPointDaemon.cc File Reference

```
#include "tpcDewPointSender.hh"  
#include <unistd.h>
```

Functions

- void [runSender](#) (const char *lDir)

5.6.1 Function Documentation

5.6.1.1 void runSender (const char * lDir)

Definition at line 14 of file tpcDewPointDaemon.cc.

```
14         {  
15  
16     CndDbSender* sender = new tpcDewPointSender(lDir);  
17  
18     sender->initQuery();  
19     for(;;) { //ever...  
20         if(sender->hasBackLog())sender->cleanBackLog();  
21         if(sender->queryData())sender->updateDb();  
22         sleep(sender->sleepTime());  
23     }  
24  
25 };
```

5.7 tpcDewPointSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include "tpcDewPointSender.hh"
#include "StDbTable.h"
#include "tpcDewPointSender_i.cc"
```

Defines

- `#define __CLASS__ "tpcDewPointSender"`
- `#define __METHOD__ "initTable()"`
- `#define __METHOD__ "initDataBase()"`
- `#define __METHOD__ "queryData()"`
- `#define __METHOD__ "readData(fileName)"`
- `#define __METHOD__ "updateDb(filename)"`

5.7.1 Define Documentation

5.7.1.1 `#define __CLASS__ "tpcDewPointSender"`

Definition at line 17 of file tpcDewPointSender.cc.

5.7.1.2 `#define __METHOD__ "updateDb(filename)"`

5.7.1.3 `#define __METHOD__ "readData(fileName)"`

5.7.1.4 `#define __METHOD__ "queryData()"`

5.7.1.5 `#define __METHOD__ "initDataBase()"`

5.7.1.6 `#define __METHOD__ "initTable()"`

5.8 tpcDewPointSender.hh File Reference

```
#include "CndDbSender.hh"  
#include "tpcDewPoint.h"
```

Classes

- class [tpcDewPointSender](#)

Defines

- #define [NUM_DB_ROWS](#) 1

5.8.1 Define Documentation

5.8.1.1 #define NUM_DB_ROWS 1

Definition at line 16 of file tpcDewPointSender.hh.

5.9 tpcDewPointSender_i.cc File Reference

Defines

- #define `__METHOD__` "loadUserControls(name,value)"
- #define `__METHOD__` "initQuery()"
- #define `__METHOD__` "readData(ifstream)"

5.9.1 Define Documentation

5.9.1.1 #define `__METHOD__` "readData(ifstream)"

5.9.1.2 #define `__METHOD__` "initQuery()"

5.9.1.3 #define `__METHOD__` "loadUserControls(name,value)"

5.10 tpcFieldCageDaemon.cc File Reference

```
#include "tpcFieldCageSender.hh"  
#include <unistd.h>
```

Functions

- void [runSender](#) (const char *ldir)

5.10.1 Function Documentation

5.10.1.1 void runSender (const char * ldir)

Definition at line 14 of file tpcFieldCageDaemon.cc.

```
14         {  
15  
16     CndDbSender* sender = new tpcFieldCageSender(ldir);  
17  
18     sender->initQuery();  
19     for(;;) { //ever...  
20         if(sender->hasBackLog())sender->cleanBackLog();  
21         if(sender->queryData())sender->updateDb();  
22         sleep(sender->sleepTime());  
23     }  
24  
25 };
```

5.11 tpcFieldCageSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include "tpcFieldCageSender.hh"
#include "StDbTable.h"
#include "tpcFieldCageSender_i.cc"
```

Defines

- `#define __CLASS__ "tpcFieldCageSender"`
- `#define __METHOD__ "initTable()"`
- `#define __METHOD__ "initDataBase()"`
- `#define __METHOD__ "queryData()"`
- `#define __METHOD__ "readData(fileName)"`
- `#define __METHOD__ "updateDb(filename)"`

5.11.1 Define Documentation

5.11.1.1 `#define __CLASS__ "tpcFieldCageSender"`

Definition at line 17 of file tpcFieldCageSender.cc.

5.11.1.2 `#define __METHOD__ "updateDb(filename)"`

5.11.1.3 `#define __METHOD__ "readData(fileName)"`

5.11.1.4 `#define __METHOD__ "queryData()"`

5.11.1.5 `#define __METHOD__ "initDataBase()"`

5.11.1.6 `#define __METHOD__ "initTable()"`

5.12 tpcFieldCageSender.hh File Reference

```
#include "CndDbSender.hh"  
#include "tpcFieldCage.h"
```

Classes

- class [tpcFieldCageSender](#)

Defines

- #define [NUM_DB_ROWS](#) 1

5.12.1 Define Documentation

5.12.1.1 #define NUM_DB_ROWS 1

Definition at line 16 of file tpcFieldCageSender.hh.

5.13 tpcFieldCageSender_i.cc File Reference

Defines

- `#define __METHOD__ "loadUserControls(name,value)"`
- `#define __METHOD__ "initQuery()"`
- `#define __METHOD__ "readData(ifstream)"`

5.13.1 Define Documentation

5.13.1.1 `#define __METHOD__ "readData(ifstream)"`

5.13.1.2 `#define __METHOD__ "initQuery()"`

5.13.1.3 `#define __METHOD__ "loadUserControls(name,value)"`

5.14 tpcGainMonitorDaemon.cc File Reference

```
#include "tpcGainMonitorSender.hh"
#include <unistd.h>
```

Functions

- void [runSender](#) (const char *lDir)

5.14.1 Function Documentation

5.14.1.1 void runSender (const char * lDir)

Definition at line 14 of file tpcGainMonitorDaemon.cc.

```
14         {
15
16     CndDbSender* sender = new tpcGainMonitorSender(lDir);
17
18     sender->initQuery();
19     for(;;) { //ever...
20         if(sender->hasBackLog())sender->cleanBackLog();
21         if(sender->queryData())sender->updateDb();
22         sleep(sender->sleepTime());
23     }
24
25 };
```

5.15 tpcGainMonitorSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include "tpcGainMonitorSender.hh"
#include "StDbTable.h"
#include "tpcGainMonitorSender_i.cc"
```

Defines

- `#define __CLASS__ "tpcGainMonitorSender"`
- `#define __METHOD__ "initTable()"`
- `#define __METHOD__ "initDataBase()"`
- `#define __METHOD__ "queryData()"`
- `#define __METHOD__ "readData(fileName)"`
- `#define __METHOD__ "updateDb(filename)"`

5.15.1 Define Documentation

5.15.1.1 `#define __CLASS__ "tpcGainMonitorSender"`

Definition at line 17 of file tpcGainMonitorSender.cc.

5.15.1.2 `#define __METHOD__ "updateDb(filename)"`

5.15.1.3 `#define __METHOD__ "readData(fileName)"`

5.15.1.4 `#define __METHOD__ "queryData()"`

5.15.1.5 `#define __METHOD__ "initDataBase()"`

5.15.1.6 `#define __METHOD__ "initTable()"`

5.16 tpcGainMonitorSender.hh File Reference

```
#include "CndDbSender.hh"  
#include "tpcGainMonitor.h"
```

Classes

- class [tpcGainMonitorSender](#)

Defines

- #define [NUM_DB_ROWS](#) 1

5.16.1 Define Documentation

5.16.1.1 #define NUM_DB_ROWS 1

Definition at line 16 of file tpcGainMonitorSender.hh.

5.17 tpcGainMonitorSender_i.cc File Reference

Defines

- #define `__METHOD__` "loadUserControls(name,value)"
- #define `__METHOD__` "initQuery()"
- #define `__METHOD__` "readData(ifstream)"

5.17.1 Define Documentation

5.17.1.1 #define `__METHOD__` "readData(ifstream)"

5.17.1.2 #define `__METHOD__` "initQuery()"

5.17.1.3 #define `__METHOD__` "loadUserControls(name,value)"

5.18 tpcGasDaemon.cc File Reference

```
#include "tpcGasSender.hh"  
#include <unistd.h>
```

Functions

- void [runSender](#) (const char *ldir)

5.18.1 Function Documentation

5.18.1.1 void runSender (const char * ldir)

Definition at line 14 of file tpcGasDaemon.cc.

```
14         {  
15  
16     CndDbSender* sender = new tpcGasSender(ldir);  
17  
18     sender->initQuery();  
19     for(;;) { //ever...  
20         if(sender->hasBackLog())sender->cleanBackLog();  
21         if(sender->queryData())sender->updateDb();  
22         sleep(sender->sleepTime());  
23     }  
24  
25 };
```

5.19 tpcGasSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include "tpcGasSender.hh"
#include "StDbTable.h"
#include "tpcGasSender_i.cc"
```

Defines

- #define `__CLASS__` "tpcGasSender"
- #define `__METHOD__` "initTable()"
- #define `__METHOD__` "initDataBase()"
- #define `__METHOD__` "queryData()"
- #define `__METHOD__` "readData(fileName)"
- #define `__METHOD__` "updateDb(filename)"

5.19.1 Define Documentation

5.19.1.1 #define `__CLASS__` "tpcGasSender"

Definition at line 16 of file tpcGasSender.cc.

5.19.1.2 #define `__METHOD__` "updateDb(filename)"

5.19.1.3 #define `__METHOD__` "readData(fileName)"

5.19.1.4 #define `__METHOD__` "queryData()"

5.19.1.5 #define `__METHOD__` "initDataBase()"

5.19.1.6 #define `__METHOD__` "initTable()"

5.20 tpcGasSender.hh File Reference

```
#include "CndDbSender.hh"  
#include "tpcGas.h"
```

Classes

- class [tpcGasSender](#)

Defines

- #define [NUM_DB_ROWS](#) 1

5.20.1 Define Documentation

5.20.1.1 #define NUM_DB_ROWS 1

Definition at line 16 of file tpcGasSender.hh.

5.21 tpcGasSender_i.cc File Reference

Defines

- `#define __METHOD__ "loadUserControls(name,value)"`
- `#define __METHOD__ "initQuery()"`
- `#define __METHOD__ "readData(ifstream)"`

5.21.1 Define Documentation

5.21.1.1 `#define __METHOD__ "readData(ifstream)"`

5.21.1.2 `#define __METHOD__ "initQuery()"`

5.21.1.3 `#define __METHOD__ "loadUserControls(name,value)"`

5.22 tpcISANodeDaemon.cc File Reference

```
#include "tpcISANodeSender.hh"  
#include <unistd.h>
```

Functions

- void [runSender](#) (const char *ldir)

5.22.1 Function Documentation

5.22.1.1 void runSender (const char * ldir)

Definition at line 14 of file tpcISANodeDaemon.cc.

```
14         {  
15  
16     CndDbSender* sender = new tpcISANodeSender(ldir);  
17  
18     sender->initQuery();  
19     for(;;) { //ever...  
20         if(sender->hasBackLog())sender->cleanBackLog();  
21         if(sender->queryData())sender->updateDb();  
22         sleep(sender->sleepTime());  
23     }  
24  
25 };
```

5.23 tpcISANodeSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include "tpcISANodeSender.hh"
#include "StDbTable.h"
#include "tpcISANodeSender_i.cc"
```

Defines

- `#define __CLASS__ "tpcISANodeSender"`
- `#define __METHOD__ "initTable()"`
- `#define __METHOD__ "initDataBase()"`
- `#define __METHOD__ "queryData()"`
- `#define __METHOD__ "readData(fileName)"`
- `#define __METHOD__ "updateDb(filename)"`

5.23.1 Define Documentation

5.23.1.1 `#define __CLASS__ "tpcISANodeSender"`

Definition at line 17 of file tpcISANodeSender.cc.

5.23.1.2 `#define __METHOD__ "updateDb(filename)"`

5.23.1.3 `#define __METHOD__ "readData(fileName)"`

5.23.1.4 `#define __METHOD__ "queryData()"`

5.23.1.5 `#define __METHOD__ "initDataBase()"`

5.23.1.6 `#define __METHOD__ "initTable()"`

5.24 tpcISANodeSender.hh File Reference

```
#include "CndDbSender.hh"  
#include "tpcISANode.h"
```

Classes

- class [tpcISANodeSender](#)

Defines

- #define [NUM_DB_ROWS](#) 24

5.24.1 Define Documentation

5.24.1.1 #define NUM_DB_ROWS 24

Definition at line 16 of file tpcISANodeSender.hh.

5.25 tpcISANodeSender_i.cc File Reference

Defines

- #define `__METHOD__` "loadUserControls(name,value)"
- #define `__METHOD__` "initQuery()"
- #define `__METHOD__` "readData(ifstream)"

5.25.1 Define Documentation

5.25.1.1 #define `__METHOD__` "readData(ifstream)"

5.25.1.2 #define `__METHOD__` "initQuery()"

5.25.1.3 #define `__METHOD__` "loadUserControls(name,value)"

5.26 tpcISGGridDaemon.cc File Reference

```
#include "tpcISGGridSender.hh"  
#include <unistd.h>
```

Functions

- void [runSender](#) (const char *ldir)

5.26.1 Function Documentation

5.26.1.1 void runSender (const char * ldir)

Definition at line 14 of file tpcISGGridDaemon.cc.

```
14         {  
15  
16     CndDbSender* sender = new tpcISGGridSender(ldir);  
17  
18     sender->initQuery();  
19     for(;;) { //ever...  
20         if(sender->hasBackLog())sender->cleanBackLog();  
21         if(sender->queryData())sender->updateDb();  
22         sleep(sender->sleepTime());  
23     }  
24  
25 };
```

5.27 tpcISGGridSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include "tpcISGGridSender.hh"
#include "StDbTable.h"
#include "tpcISGGridSender_i.cc"
```

Defines

- #define `__CLASS__` "tpcISGGridSender"
- #define `__METHOD__` "initTable()"
- #define `__METHOD__` "initDataBase()"
- #define `__METHOD__` "queryData()"
- #define `__METHOD__` "readData(fileName)"
- #define `__METHOD__` "updateDb(filename)"

5.27.1 Define Documentation

5.27.1.1 #define `__CLASS__` "tpcISGGridSender"

Definition at line 17 of file tpcISGGridSender.cc.

5.27.1.2 #define `__METHOD__` "updateDb(filename)"

5.27.1.3 #define `__METHOD__` "readData(fileName)"

5.27.1.4 #define `__METHOD__` "queryData()"

5.27.1.5 #define `__METHOD__` "initDataBase()"

5.27.1.6 #define `__METHOD__` "initTable()"

5.28 tpcISGGridSender.hh File Reference

```
#include "CndDbSender.hh"  
#include "tpcISGGrid.h"
```

Classes

- class [tpcISGGridSender](#)

Defines

- #define [NUM_DB_ROWS](#) 24

5.28.1 Define Documentation

5.28.1.1 #define NUM_DB_ROWS 24

Definition at line 16 of file tpcISGGridSender.hh.

5.29 tpcISGGridSender_i.cc File Reference

Defines

- `#define __METHOD__ "loadUserControls(name,value)"`
- `#define __METHOD__ "initQuery()"`
- `#define __METHOD__ "readData(ifstream)"`

5.29.1 Define Documentation

5.29.1.1 `#define __METHOD__ "readData(ifstream)"`

5.29.1.2 `#define __METHOD__ "initQuery()"`

5.29.1.3 `#define __METHOD__ "loadUserControls(name,value)"`

5.30 tpcOSANodeDaemon.cc File Reference

```
#include "tpcOSANodeSender.hh"  
#include <unistd.h>
```

Functions

- void [runSender](#) (const char *ldir)

5.30.1 Function Documentation

5.30.1.1 void runSender (const char * ldir)

Definition at line 14 of file tpcOSANodeDaemon.cc.

```
14         {  
15  
16     CndDbSender* sender = new tpcOSANodeSender(ldir);  
17  
18     sender->initQuery();  
19     for(;;) { //ever...  
20         if(sender->hasBackLog())sender->cleanBackLog();  
21         if(sender->queryData())sender->updateDb();  
22         sleep(sender->sleepTime());  
23     }  
24  
25 };
```

5.31 tpcOSANodeSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include "tpcOSANodeSender.hh"
#include "StDbTable.h"
#include "tpcOSANodeSender_i.cc"
```

Defines

- #define `__CLASS__` "tpcOSANodeSender"
- #define `__METHOD__` "initTable()"
- #define `__METHOD__` "initDataBase()"
- #define `__METHOD__` "queryData()"
- #define `__METHOD__` "readData(fileName)"
- #define `__METHOD__` "updateDb(filename)"

5.31.1 Define Documentation

5.31.1.1 #define `__CLASS__` "tpcOSANodeSender"

Definition at line 17 of file tpcOSANodeSender.cc.

5.31.1.2 #define `__METHOD__` "updateDb(filename)"

5.31.1.3 #define `__METHOD__` "readData(fileName)"

5.31.1.4 #define `__METHOD__` "queryData()"

5.31.1.5 #define `__METHOD__` "initDataBase()"

5.31.1.6 #define `__METHOD__` "initTable()"

5.32 tpcOSANodeSender.hh File Reference

```
#include "CndDbSender.hh"  
#include "tpcOSANode.h"
```

Classes

- class [tpcOSANodeSender](#)

Defines

- #define [NUM_DB_ROWS](#) 24

5.32.1 Define Documentation

5.32.1.1 #define NUM_DB_ROWS 24

Definition at line 16 of file tpcOSANodeSender.hh.

5.33 tpcOSANodeSender_i.cc File Reference

Defines

- #define `__METHOD__` "loadUserControls(name,value)"
- #define `__METHOD__` "initQuery()"
- #define `__METHOD__` "readData(ifstream)"

5.33.1 Define Documentation

5.33.1.1 #define `__METHOD__` "readData(ifstream)"

5.33.1.2 #define `__METHOD__` "initQuery()"

5.33.1.3 #define `__METHOD__` "loadUserControls(name,value)"

5.34 tpcOSGGridDaemon.cc File Reference

```
#include "tpcOSGGridSender.hh"  
#include <unistd.h>
```

Functions

- void [runSender](#) (const char *ldir)

5.34.1 Function Documentation

5.34.1.1 void runSender (const char * ldir)

Definition at line 14 of file tpcOSGGridDaemon.cc.

```
14         {  
15  
16     CndDbSender* sender = new tpcOSGGridSender(ldir);  
17  
18     sender->initQuery();  
19     for(;;) { //ever...  
20         if(sender->hasBackLog())sender->cleanBackLog();  
21         if(sender->queryData())sender->updateDb();  
22         sleep(sender->sleepTime());  
23     }  
24  
25 };
```

5.35 tpcOSGGridSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include "tpcOSGGridSender.hh"
#include "StDbTable.h"
#include "tpcOSGGridSender_i.cc"
```

Defines

- `#define __CLASS__ "tpcOSGGridSender"`
- `#define __METHOD__ "initTable()"`
- `#define __METHOD__ "initDataBase()"`
- `#define __METHOD__ "queryData()"`
- `#define __METHOD__ "readData(fileName)"`
- `#define __METHOD__ "updateDb(filename)"`

5.35.1 Define Documentation

5.35.1.1 `#define __CLASS__ "tpcOSGGridSender"`

Definition at line 17 of file tpcOSGGridSender.cc.

5.35.1.2 `#define __METHOD__ "updateDb(filename)"`

5.35.1.3 `#define __METHOD__ "readData(fileName)"`

5.35.1.4 `#define __METHOD__ "queryData()"`

5.35.1.5 `#define __METHOD__ "initDataBase()"`

5.35.1.6 `#define __METHOD__ "initTable()"`

5.36 tpcOSGGridSender.hh File Reference

```
#include "CndDbSender.hh"  
#include "tpcOSGGrid.h"
```

Classes

- class [tpcOSGGridSender](#)

Defines

- #define [NUM_DB_ROWS](#) 24

5.36.1 Define Documentation

5.36.1.1 #define NUM_DB_ROWS 24

Definition at line 16 of file tpcOSGGridSender.hh.

5.37 tpcOSGGridSender_i.cc File Reference

Defines

- #define `__METHOD__` "loadUserControls(name,value)"
- #define `__METHOD__` "initQuery()"
- #define `__METHOD__` "readData(ifstream)"

5.37.1 Define Documentation

5.37.1.1 #define `__METHOD__` "readData(ifstream)"

5.37.1.2 #define `__METHOD__` "initQuery()"

5.37.1.3 #define `__METHOD__` "loadUserControls(name,value)"

Index

- ~tpcAnodeStateSender
 - tpcAnodeStateSender, 8
- ~tpcAnodeStateSender_i
 - tpcAnodeStateSender_i, 12
- ~tpcDewPointSender
 - tpcDewPointSender, 17
- ~tpcFieldCageSender
 - tpcFieldCageSender, 27
- ~tpcGainMonitorSender
 - tpcGainMonitorSender, 38
- ~tpcGasSender
 - tpcGasSender, 48
- ~tpcISAnodeSender
 - tpcISAnodeSender, 59
- ~tpcISGGridSender
 - tpcISGGridSender, 70
- ~tpcOSAnodeSender
 - tpcOSAnodeSender, 81
- ~tpcOSGGridSender
 - tpcOSGGridSender, 92
- __CLASS__
 - tpcAnodeStateSender.cc, 104
 - tpcAnodeStateSender_i.cc, 106
 - tpcDewPointSender.cc, 109
 - tpcFieldCageSender.cc, 113
 - tpcGainMonitorSender.cc, 117
 - tpcGasSender.cc, 121
 - tpcISAnodeSender.cc, 125
 - tpcISGGridSender.cc, 129
 - tpcOSAnodeSender.cc, 133
 - tpcOSGGridSender.cc, 137
- __METHOD__
 - tpcAnodeStateSender.cc, 104
 - tpcAnodeStateSender_i.cc, 106
 - tpcDewPointSender.cc, 109
 - tpcDewPointSender_i.cc, 111
 - tpcFieldCageSender.cc, 113
 - tpcFieldCageSender_i.cc, 115
 - tpcGainMonitorSender.cc, 117
 - tpcGainMonitorSender_i.cc, 119
 - tpcGasSender.cc, 121
 - tpcGasSender_i.cc, 123
 - tpcISAnodeSender.cc, 125
 - tpcISAnodeSender_i.cc, 127
 - tpcISGGridSender.cc, 129
 - tpcISGGridSender_i.cc, 131
 - tpcOSAnodeSender.cc, 133
 - tpcOSAnodeSender_i.cc, 135
 - tpcOSGGridSender.cc, 137
 - tpcOSGGridSender_i.cc, 139
- cdriftLimit
 - tpcFieldCageSender, 35
 - tpcISAnodeSender, 67
 - tpcOSAnodeSender, 89
- elementList
 - tpcAnodeStateSender, 10
 - tpcDewPointSender, 24
 - tpcFieldCageSender, 35
 - tpcGainMonitorSender, 45
 - tpcGasSender, 56
 - tpcISAnodeSender, 67
 - tpcISGGridSender, 78
 - tpcOSAnodeSender, 89
 - tpcOSGGridSender, 100
- gsdriftLimit
 - tpcFieldCageSender, 35
- hasChanged
 - tpcAnodeStateSender, 8
 - tpcAnodeStateSender_i, 13
 - tpcDewPointSender, 17
 - tpcFieldCageSender, 27
 - tpcGainMonitorSender, 38
 - tpcGasSender, 48
 - tpcISAnodeSender, 59
 - tpcISGGridSender, 70
 - tpcOSAnodeSender, 81
 - tpcOSGGridSender, 92
- hdriftLimit
 - tpcDewPointSender, 24
- initDataBase
 - tpcAnodeStateSender, 8
 - tpcAnodeStateSender_i, 13
 - tpcDewPointSender, 17
 - tpcFieldCageSender, 27
 - tpcGainMonitorSender, 38
 - tpcGasSender, 48

- tpcISAnodeSender, 59
- tpcISGGridSender, 70
- tpcOSAnodeSender, 81
- tpcOSGGridSender, 92
- initQuery
 - tpcAnodeStateSender, 8
 - tpcAnodeStateSender_i, 13
 - tpcDewPointSender, 18
 - tpcFieldCageSender, 28
 - tpcGainMonitorSender, 39
 - tpcGasSender, 48
 - tpcISAnodeSender, 60
 - tpcISGGridSender, 71
 - tpcOSAnodeSender, 82
 - tpcOSGGridSender, 93
- initTable
 - tpcAnodeStateSender, 8
 - tpcDewPointSender, 18
 - tpcFieldCageSender, 28
 - tpcGainMonitorSender, 39
 - tpcGasSender, 50
 - tpcISAnodeSender, 60
 - tpcISGGridSender, 71
 - tpcOSAnodeSender, 82
 - tpcOSGGridSender, 93
- initTags
 - tpcAnodeStateSender, 9
 - tpcAnodeStateSender_i, 14
 - tpcDewPointSender, 19
 - tpcFieldCageSender, 29
 - tpcGainMonitorSender, 40
 - tpcGasSender, 50
 - tpcISAnodeSender, 61
 - tpcISGGridSender, 72
 - tpcOSAnodeSender, 83
 - tpcOSGGridSender, 94
- loadUserControls
 - tpcAnodeStateSender, 9
 - tpcAnodeStateSender_i, 14
 - tpcDewPointSender, 19
 - tpcFieldCageSender, 29
 - tpcGainMonitorSender, 40
 - tpcGasSender, 50
 - tpcISAnodeSender, 61
 - tpcISGGridSender, 72
 - tpcOSAnodeSender, 83
 - tpcOSGGridSender, 94
- mline
 - tpcDewPointSender, 24
 - tpcFieldCageSender, 35
 - tpcGainMonitorSender, 45
 - tpcGasSender, 56
- tpcISAnodeSender, 67
- tpcISGGridSender, 78
- tpcOSAnodeSender, 89
- tpcOSGGridSender, 100
- mreadStatus
 - tpcAnodeStateSender, 10
 - tpcDewPointSender, 24
 - tpcFieldCageSender, 35
 - tpcGainMonitorSender, 45
 - tpcGasSender, 56
 - tpcISAnodeSender, 67
 - tpcISGGridSender, 78
 - tpcOSAnodeSender, 89
 - tpcOSGGridSender, 100
- nextLine
 - tpcDewPointSender, 19
 - tpcFieldCageSender, 30
 - tpcGainMonitorSender, 40
 - tpcGasSender, 51
 - tpcISAnodeSender, 62
 - tpcISGGridSender, 73
 - tpcOSAnodeSender, 84
 - tpcOSGGridSender, 94
- NUM_DB_ROWS
 - tpcAnodeStateSender.hh, 105
 - tpcDewPointSender.hh, 110
 - tpcFieldCageSender.hh, 114
 - tpcGainMonitorSender.hh, 118
 - tpcGasSender.hh, 122
 - tpcISAnodeSender.hh, 126
 - tpcISGGridSender.hh, 130
 - tpcOSAnodeSender.hh, 134
 - tpcOSGGridSender.hh, 138
- previousVals
 - tpcAnodeStateSender, 10
 - tpcDewPointSender, 25
 - tpcFieldCageSender, 35
 - tpcGainMonitorSender, 45
 - tpcGasSender, 56
 - tpcISAnodeSender, 67
 - tpcISGGridSender, 78
 - tpcOSAnodeSender, 89
 - tpcOSGGridSender, 100
- ptr1
 - tpcDewPointSender, 25
 - tpcFieldCageSender, 35
 - tpcGainMonitorSender, 45
 - tpcGasSender, 56
 - tpcISAnodeSender, 67
 - tpcISGGridSender, 78
 - tpcOSAnodeSender, 89
 - tpcOSGGridSender, 100

- ptr2
 - tpcDewPointSender, 25
 - tpcFieldCageSender, 35
 - tpcGainMonitorSender, 45
 - tpcGasSender, 57
 - tpcISAnodeSender, 67
 - tpcISGGridSender, 78
 - tpcOSAnodeSender, 89
 - tpcOSGGridSender, 100
- queryData
 - tpcAnodeStateSender, 9
 - tpcAnodeStateSender_i, 14
 - tpcDewPointSender, 20
 - tpcFieldCageSender, 30
 - tpcGainMonitorSender, 40
 - tpcGasSender, 51
 - tpcISAnodeSender, 62
 - tpcISGGridSender, 73
 - tpcOSAnodeSender, 84
 - tpcOSGGridSender, 95
- readAny
 - tpcDewPointSender, 20
 - tpcFieldCageSender, 30
 - tpcGainMonitorSender, 41
 - tpcGasSender, 51
 - tpcISAnodeSender, 62
 - tpcISGGridSender, 73
 - tpcOSAnodeSender, 84
 - tpcOSGGridSender, 95
- readData
 - tpcAnodeStateSender, 9
 - tpcAnodeStateSender_i, 15
 - tpcDewPointSender, 20, 21
 - tpcFieldCageSender, 31
 - tpcGainMonitorSender, 41, 42
 - tpcGasSender, 52, 53
 - tpcISAnodeSender, 63
 - tpcISGGridSender, 73, 74
 - tpcOSAnodeSender, 85
 - tpcOSGGridSender, 95, 96
- readError
 - tpcDewPointSender, 21
 - tpcFieldCageSender, 32
 - tpcGainMonitorSender, 42
 - tpcGasSender, 53
 - tpcISAnodeSender, 64
 - tpcISGGridSender, 75
 - tpcOSAnodeSender, 86
 - tpcOSGGridSender, 97
- readVal
 - tpcDewPointSender, 21–23
 - tpcFieldCageSender, 32, 33
 - tpcGainMonitorSender, 42–44
 - tpcGasSender, 53–55
 - tpcISAnodeSender, 64, 65
 - tpcISGGridSender, 75, 76
 - tpcOSAnodeSender, 86, 87
 - tpcOSGGridSender, 97, 98
- runSender
 - tpcAnodeStateDaemon.cc, 103
 - tpcDewPointDaemon.cc, 108
 - tpcFieldCageDaemon.cc, 112
 - tpcGainMonitorDaemon.cc, 116
 - tpcGasDaemon.cc, 120
 - tpcISAnodeDaemon.cc, 124
 - tpcISGGridDaemon.cc, 128
 - tpcOSAnodeDaemon.cc, 132
 - tpcOSGGridDaemon.cc, 136
- tdriftLimit
 - tpcDewPointSender, 25
- tempVals
 - tpcAnodeStateSender, 11
 - tpcDewPointSender, 25
 - tpcFieldCageSender, 35
 - tpcGainMonitorSender, 45
 - tpcGasSender, 57
 - tpcISAnodeSender, 67
 - tpcISGGridSender, 78
 - tpcOSAnodeSender, 89
 - tpcOSGGridSender, 100
- tmpline
 - tpcDewPointSender, 25
 - tpcFieldCageSender, 35
 - tpcGainMonitorSender, 45
 - tpcGasSender, 57
 - tpcISAnodeSender, 67
 - tpcISGGridSender, 78
 - tpcOSAnodeSender, 89
 - tpcOSGGridSender, 100
- tpcAnodeStateDaemon.cc, 103
- tpcAnodeStateDaemon.cc
 - runSender, 103
- tpcAnodeStateSender, 7
 - tpcAnodeStateSender, 8
- tpcAnodeStateSender
 - ~tpcAnodeStateSender, 8
 - elementList, 10
 - hasChanged, 8
 - initDataBase, 8
 - initQuery, 8
 - initTable, 8
 - initTags, 9
 - loadUserControls, 9
 - mreadStatus, 10
 - previousVals, 10

- queryData, 9
- readData, 9
- tempVals, 11
- tpcAnodeStateSender, 8
- updateDb, 9
- updateElements, 11
- updateVals, 11
- tpcAnodeStateSender.cc, 104
- tpcAnodeStateSender.cc
 - __CLASS__, 104
 - __METHOD__, 104
- tpcAnodeStateSender.hh, 105
- tpcAnodeStateSender.hh
 - NUM_DB_ROWS, 105
- tpcAnodeStateSender_i, 12
 - tpcAnodeStateSender_i, 12
- tpcAnodeStateSender_i
 - ~tpcAnodeStateSender_i, 12
 - hasChanged, 13
 - initDataBase, 13
 - initQuery, 13
 - initTags, 14
 - loadUserControls, 14
 - queryData, 14
 - readData, 15
 - tpcAnodeStateSender_i, 12
- tpcAnodeStateSender_i.cc, 106
- tpcAnodeStateSender_i.cc
 - __CLASS__, 106
 - __METHOD__, 106
- tpcAnodeStateSender_i.hh, 107
- tpcDewPointDaemon.cc, 108
- tpcDewPointDaemon.cc
 - runSender, 108
- tpcDewPointSender, 16
 - tpcDewPointSender, 17
- tpcDewPointSender
 - ~tpcDewPointSender, 17
 - elementList, 24
 - hasChanged, 17
 - hdriftLimit, 24
 - initDataBase, 17
 - initQuery, 18
 - initTable, 18
 - initTags, 19
 - loadUserControls, 19
 - mline, 24
 - mreadStatus, 24
 - nextLine, 19
 - previousVals, 25
 - ptr1, 25
 - ptr2, 25
 - queryData, 20
 - readAny, 20
 - readData, 20, 21
 - readError, 21
 - readVal, 21–23
 - tdriftLimit, 25
 - tempVals, 25
 - tmpline, 25
 - tpcDewPointSender, 17
 - updateDb, 23
 - updateElements, 25
 - updateVals, 25
- tpcDewPointSender.cc, 109
- tpcDewPointSender.cc
 - __CLASS__, 109
 - __METHOD__, 109
- tpcDewPointSender.hh, 110
- tpcDewPointSender.hh
 - NUM_DB_ROWS, 110
- tpcDewPointSender_i.cc, 111
- tpcDewPointSender_i.cc
 - __METHOD__, 111
- tpcFieldCageDaemon.cc, 112
- tpcFieldCageDaemon.cc
 - runSender, 112
- tpcFieldCageSender, 26
 - tpcFieldCageSender, 27
- tpcFieldCageSender
 - ~tpcFieldCageSender, 27
 - cdriftLimit, 35
 - elementList, 35
 - gscdriftLimit, 35
 - hasChanged, 27
 - initDataBase, 27
 - initQuery, 28
 - initTable, 28
 - initTags, 29
 - loadUserControls, 29
 - mline, 35
 - mreadStatus, 35
 - nextLine, 30
 - previousVals, 35
 - ptr1, 35
 - ptr2, 35
 - queryData, 30
 - readAny, 30
 - readData, 31
 - readError, 32
 - readVal, 32, 33
 - tempVals, 35
 - tmpline, 35
 - tpcFieldCageSender, 27
 - updateDb, 34
 - updateElements, 35
 - updateVals, 36
 - vdriфтLimit, 36

- tpcFieldCageSender.cc, 113
- tpcFieldCageSender.cc
 - __CLASS__, 113
 - __METHOD__, 113
- tpcFieldCageSender.hh, 114
- tpcFieldCageSender.hh
 - NUM_DB_ROWS, 114
- tpcFieldCageSender_i.cc, 115
- tpcFieldCageSender_i.cc
 - __METHOD__, 115
- tpcGainMonitorDaemon.cc, 116
- tpcGainMonitorDaemon.cc
 - runSender, 116
- tpcGainMonitorSender, 37
- tpcGainMonitorSender, 38
- tpcGainMonitorSender
 - ~tpcGainMonitorSender, 38
 - elementList, 45
 - hasChanged, 38
 - initDataBase, 38
 - initQuery, 39
 - initTable, 39
 - initTags, 40
 - loadUserControls, 40
 - mline, 45
 - mreadStatus, 45
 - nextLine, 40
 - previousVals, 45
 - ptr1, 45
 - ptr2, 45
 - queryData, 40
 - readAny, 41
 - readData, 41, 42
 - readError, 42
 - readVal, 42–44
 - tempVals, 45
 - tmline, 45
 - tpcGainMonitorSender, 38
 - updateDb, 44
 - updateElements, 46
 - updateVals, 46
 - widthLimit, 46
- tpcGainMonitorSender.cc, 117
- tpcGainMonitorSender.cc
 - __CLASS__, 117
 - __METHOD__, 117
- tpcGainMonitorSender.hh, 118
- tpcGainMonitorSender.hh
 - NUM_DB_ROWS, 118
- tpcGainMonitorSender_i.cc, 119
- tpcGainMonitorSender_i.cc
 - __METHOD__, 119
- tpcGasDaemon.cc, 120
- tpcGasDaemon.cc
 - runSender, 120
- tpcGasSender, 47
- tpcGasSender, 48
- tpcGasSender
 - ~tpcGasSender, 48
 - elementList, 56
 - hasChanged, 48
 - initDataBase, 48
 - initQuery, 48
 - initTable, 50
 - initTags, 50
 - loadUserControls, 50
 - mline, 56
 - mreadStatus, 56
 - nextLine, 51
 - previousVals, 56
 - ptr1, 56
 - ptr2, 57
 - queryData, 51
 - readAny, 51
 - readData, 52, 53
 - readError, 53
 - readVal, 53–55
 - tempVals, 57
 - tmline, 57
 - tpcGasSender, 48
 - updateDb, 55
 - updateElements, 57
 - updateVals, 57
- tpcGasSender.cc, 121
- tpcGasSender.cc
 - __CLASS__, 121
 - __METHOD__, 121
- tpcGasSender.hh, 122
- tpcGasSender.hh
 - NUM_DB_ROWS, 122
- tpcGasSender_i.cc, 123
- tpcGasSender_i.cc
 - __METHOD__, 123
- tpcISANodeDaemon.cc, 124
- tpcISANodeDaemon.cc
 - runSender, 124
- tpcISANodeSender, 58
- tpcISANodeSender, 59
- tpcISANodeSender
 - ~tpcISANodeSender, 59
 - cdriftLimit, 67
 - elementList, 67
 - hasChanged, 59
 - initDataBase, 59
 - initQuery, 60
 - initTable, 60
 - initTags, 61
 - loadUserControls, 61

- mline, 67
- mreadStatus, 67
- nextLine, 62
- previousVals, 67
- ptr1, 67
- ptr2, 67
- queryData, 62
- readAny, 62
- readData, 63
- readError, 64
- readVal, 64, 65
- tempVals, 67
- tmline, 67
- tpcISANodeSender, 59
- updateDb, 66
- updateElements, 67
- updateVals, 67
- vdriftLimit, 68
- tpcISANodeSender.cc, 125
- tpcISANodeSender.cc
 - __CLASS__, 125
 - __METHOD__, 125
- tpcISANodeSender.hh, 126
- tpcISANodeSender.hh
 - NUM_DB_ROWS, 126
- tpcISANodeSender_i.cc, 127
- tpcISANodeSender_i.cc
 - __METHOD__, 127
- tpcISGGridDaemon.cc, 128
- tpcISGGridDaemon.cc
 - runSender, 128
- tpcISGGridSender, 69
- tpcISGGridSender, 70
- tpcISGGridSender
 - ~tpcISGGridSender, 70
 - elementList, 78
 - hasChanged, 70
 - initDataBase, 70
 - initQuery, 71
 - initTable, 71
 - initTags, 72
 - loadUserControls, 72
 - mline, 78
 - mreadStatus, 78
 - nextLine, 73
 - previousVals, 78
 - ptr1, 78
 - ptr2, 78
 - queryData, 73
 - readAny, 73
 - readData, 73, 74
 - readError, 75
 - readVal, 75, 76
 - tempVals, 78
 - tmline, 78
 - tpcISGGridSender, 70
 - updateDb, 77
 - updateElements, 78
 - updateVals, 78
 - vdriftLimit, 78
- tpcISGGridSender.cc, 129
- tpcISGGridSender.cc
 - __CLASS__, 129
 - __METHOD__, 129
- tpcISGGridSender.hh, 130
- tpcISGGridSender.hh
 - NUM_DB_ROWS, 130
- tpcISGGridSender_i.cc, 131
- tpcISGGridSender_i.cc
 - __METHOD__, 131
- tpcOSANodeDaemon.cc, 132
- tpcOSANodeDaemon.cc
 - runSender, 132
- tpcOSANodeSender, 80
- tpcOSANodeSender, 81
- tpcOSANodeSender
 - ~tpcOSANodeSender, 81
 - cdriftLimit, 89
 - elementList, 89
 - hasChanged, 81
 - initDataBase, 81
 - initQuery, 82
 - initTable, 82
 - initTags, 83
 - loadUserControls, 83
 - mline, 89
 - mreadStatus, 89
 - nextLine, 84
 - previousVals, 89
 - ptr1, 89
 - ptr2, 89
 - queryData, 84
 - readAny, 84
 - readData, 85
 - readError, 86
 - readVal, 86, 87
 - tempVals, 89
 - tmline, 89
 - tpcOSANodeSender, 81
 - updateDb, 88
 - updateElements, 89
 - updateVals, 89
 - vdriftLimit, 90
- tpcOSANodeSender.cc, 133
- tpcOSANodeSender.cc
 - __CLASS__, 133
 - __METHOD__, 133
- tpcOSANodeSender.hh, 134

- tpcOSAnodeSender.hh
 - NUM_DB_ROWS, 134
- tpcOSAnodeSender_i.cc, 135
- tpcOSAnodeSender_i.cc
 - __METHOD__, 135
- tpcOSGGridDaemon.cc, 136
- tpcOSGGridDaemon.cc
 - runSender, 136
- tpcOSGGridSender, 91
 - tpcOSGGridSender, 92
- tpcOSGGridSender
 - ~tpcOSGGridSender, 92
 - elementList, 100
 - hasChanged, 92
 - initDataBase, 92
 - initQuery, 93
 - initTable, 93
 - initTags, 94
 - loadUserControls, 94
 - mline, 100
 - mreadStatus, 100
 - nextLine, 94
 - previousVals, 100
 - ptr1, 100
 - ptr2, 100
 - queryData, 95
 - readAny, 95
 - readData, 95, 96
 - readError, 97
 - readVal, 97, 98
 - tempVals, 100
 - tmline, 100
 - tpcOSGGridSender, 92
 - updateDb, 99
 - updateElements, 100
 - updateVals, 100
 - vdriftLimit, 100
- tpcOSGGridSender.cc, 137
- tpcOSGGridSender.cc
 - __CLASS__, 137
 - __METHOD__, 137
- tpcOSGGridSender.hh, 138
- tpcOSGGridSender.hh
 - NUM_DB_ROWS, 138
- tpcOSGGridSender_i.cc, 139
- tpcOSGGridSender_i.cc
 - __METHOD__, 139
- updateDb
 - tpcAnodeStateSender, 9
 - tpcDewPointSender, 23
 - tpcFieldCageSender, 34
 - tpcGainMonitorSender, 44
 - tpcGasSender, 55
 - tpcISAnodeSender, 66
 - tpcISGGridSender, 77
 - tpcOSAnodeSender, 88
 - tpcOSGGridSender, 99
- updateElements
 - tpcAnodeStateSender, 11
 - tpcDewPointSender, 25
 - tpcFieldCageSender, 35
 - tpcGainMonitorSender, 46
 - tpcGasSender, 57
 - tpcISAnodeSender, 67
 - tpcISGGridSender, 78
 - tpcOSAnodeSender, 89
 - tpcOSGGridSender, 100
- updateVals
 - tpcAnodeStateSender, 11
 - tpcDewPointSender, 25
 - tpcFieldCageSender, 36
 - tpcGainMonitorSender, 46
 - tpcGasSender, 57
 - tpcISAnodeSender, 67
 - tpcISGGridSender, 78
 - tpcOSAnodeSender, 89
 - tpcOSGGridSender, 100
- vdriftLimit
 - tpcFieldCageSender, 36
 - tpcISAnodeSender, 68
 - tpcISGGridSender, 78
 - tpcOSAnodeSender, 90
 - tpcOSGGridSender, 100
- widthLimit
 - tpcGainMonitorSender, 46